Freescale Semiconductor
Application Note

Document Number: AN4131
Rev. 0, 05/2010

Getting Started with OpenGL ES, PVR Libraries, and Tools for the i.MX31 PDK

Setup for Win CE™ and Microsoft® Tools

by Multimedia Applications Division
Freescale Semiconductor, Inc.
Austin, TX

Contents

1 Introduction

3D graphics are widely used in many embedded system applications such as browser interfaces, virtual desktops, and file viewers. 3D graphics are also used in complicated 3D applications such as video games, Global Positioning Systems (GPS), and terrain generators. This application note gives an overview of the MBX R-S 3D Graphics Core and the procedure to set up the i.MX31 PDK for building 3D applications in the WinCE platform.

This application note also provides a series of steps to configure eMbedded Visual C++. This helps to create new 3D applications that work with Open Graphics Library for Embedded Systems (OpenGL ES), PowerVR (PVR) shell, and OpenGL ES (OGLES) tools. OpenGL is the language used by most of the graphics hardware. PVR shell and OGLES are a series of libraries and tools that aid the development of 3D applications.

1.1 MBX Module Overview

The MBX R-S 3D Graphics Core is an Advanced Microcontroller Bus Architecture (AMBA) compliant

System-on-a-Chip (SoC) component. Figure 1 shows a top-level block diagram of the MBX R-S 3D Graphics Core.

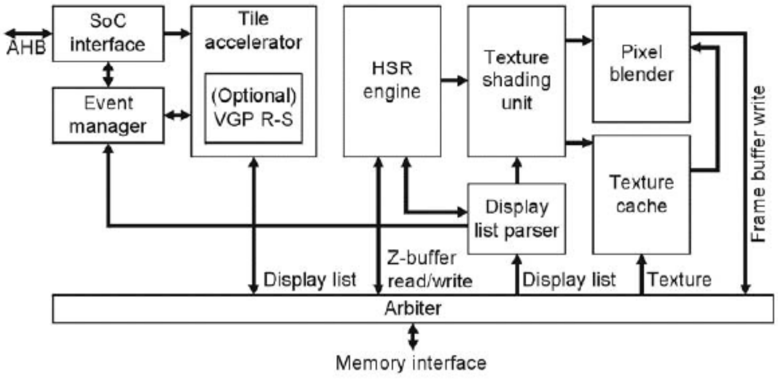


Figure 1. MBX R-S Graphics Core

The MBX R-S 3D Graphics Core consists of the following modules:

- Tile Accelerator (TA)
- Event manager
- Display list parser
- Hidden Surface Removal (HSR) engine
- Texture shading unit
- Texture cache
- Pixel blender

The MBX R-S 3D Graphics Core operates on 3D scene data (sent as triangle batches) that are transformed and lit either by the Central Processing Unit (CPU) or by the optional VGP R-S. Triangles are written directly to the TA on First In First Out (FIFO) basis, so that the CPU is not stalled. The TA performs advanced culling on triangle data by writing the tiled non-culled triangles to the external memory.

The HSR engine reads the tiled data and implements per-pixel HSR with full Z-accuracy. The resulting visible pixels are textured and shaded in Internal True Color (ITC) before rendering the final image for the display.

1.2 MBX R-S 3D Graphics Core Features

The MBX R-S 3D Graphics Core has the following features:

- Deferred texturing
- Screen tiling
- Flat and Gouraud shading
- Perspective correct texturing
- Specular highlights
- Floating-point Z-buffer

- 32-bit Alpha, Red, Green, and Blue (ARGB) internal rendering and layer buffering
- Full tile blend buffer
- Z-load and store mode
- Per-vertex fog
- 16-bit RGB textures, 1555, 565, 4444, 8332, 88
- 32-bit RGB textures, 8888
- YUV 422 textures
- PowerVR Texture Compression (PVR-TC) compressed textures
- One-bit textures for text acceleration
- Point, bilinear, trilinear, and anisotropic filtering
- Full range of OpenGL and Direct3D (D3D) blend modes
- Dot3 bump mapping
- Alpha test
- Zero-cost full-scene anti-aliasing
- 2Dvia3D

NOTE

The MBX module is present in the i.MX31 processor, but not in the i.MX31L.

2 Creating New SDK Containing MBX and VPU Functionality in WinCE Platform Builder

The prerequisites to create SDK in the WinCE platform builder are as follows:

- To complete the steps mentioned in this section, the platform builder and a proper i.MX Board Support Package (BSP) must be installed. If none of these can be installed, contact Microsoft or Freescale distributor.
- While creating a new platform in the platform builder:
  — Add the Vector Processing Unit (VPU) drivers.
  To download the VPU drivers, go to http://www.arm.com/products/os/windowsce_downloads.html, and follow the instructions provided there.

Steps to create a new Software Development Kit (SDK) in the WinCE platform builder are as follows:

1. Open Third Party > BSP > Freescale 3DS:ARM4I > Device Drivers > MBX in the WinCE catalog, and add all the catalog items in the MBX branch to the OS Design, as shown in Figure 2.

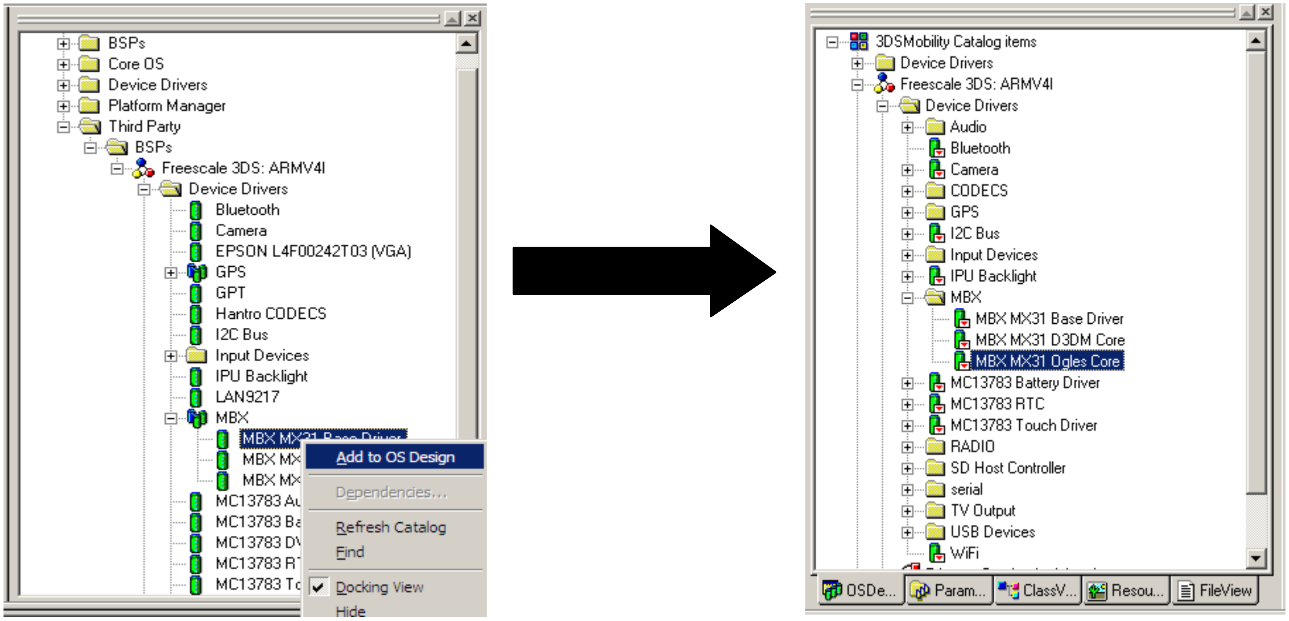


Figure 2. Catalog Items

2. Select SDK > New SDK in the Platform menu, and follow the SDK Wizard to configure the SDK, as shown in Figure 3.

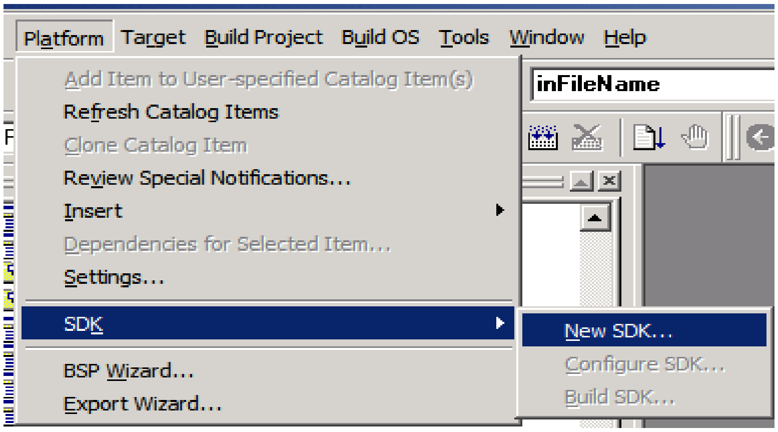
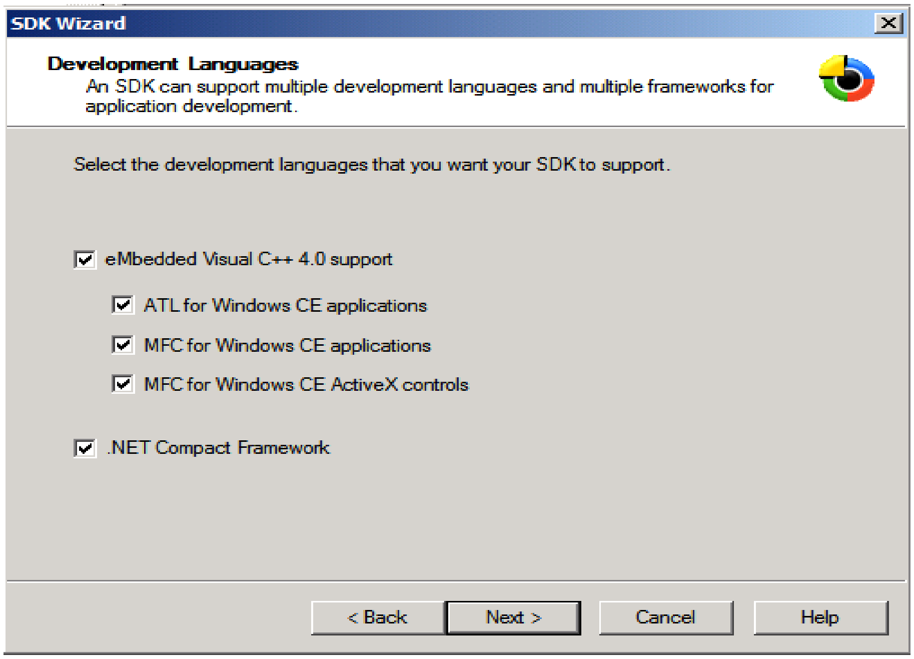


Figure 3. Platform Menu

3. Select the settings as shown in Figure 4, and click Next to complete the SDK Wizard.



Development Languages

An SDK can support multiple development languages and multiple frameworks for application development.

Select the development languages that you want your SDK to support.

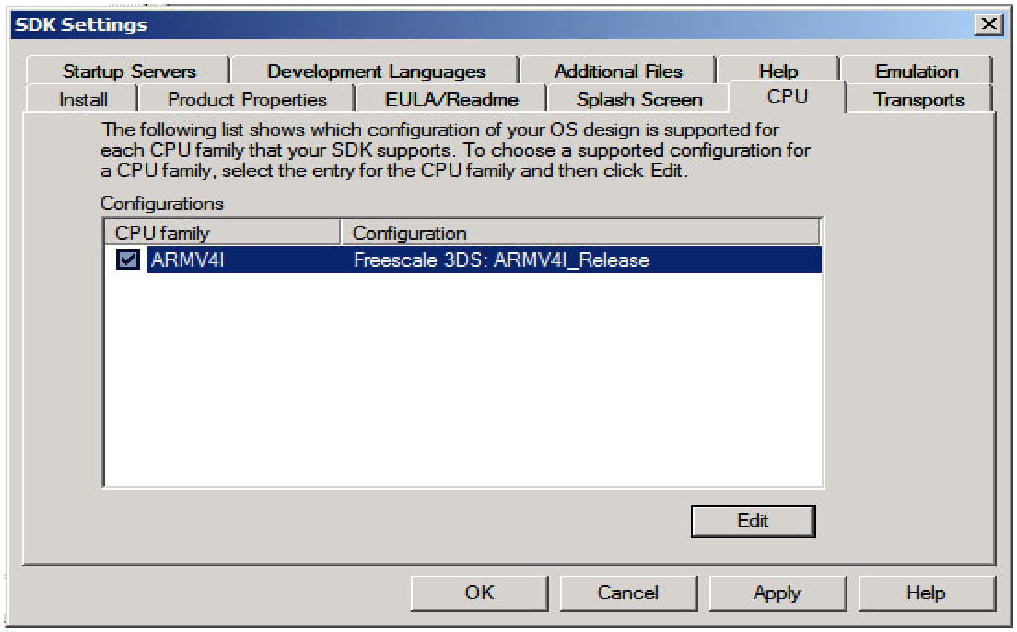
☑ eMbedded Visual C++ 4.0 support

- ☑ ATL for Windows CE applications
- ☑ MFC for Windows CE applications
- ☑ MFC for Windows CE ActiveX controls

☑ .NET Compact Framework

Figure 4. SDK Wizard

4. Once the SDK Wizard completes, go to SDK > Configure SDK in the Platform menu. In the CPU tab, make sure that the configuration chosen for ARM4I is Freescale 3DS: ARMV41_Release/Debug, and then click Apply followed by OK. The SDK Settings window is shown in Figure 5.



The following list shows which configuration of your OS design is supported for each CPU family that your SDK supports. To choose a supported configuration for a CPU family, select the entry for the CPU family and then click Edit.

Configurations

CPU family	Configuration
☑ ARMV4I	Freescale 3DS: ARMV4I_Release

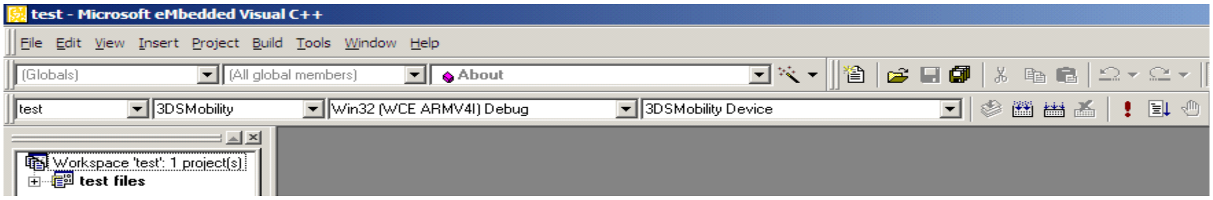
Figure 5. SDK Settings

5. Once the settings are configured, select SDK > Build SDK option in the Platform menu and follow the Wizard. When the Wizard completes, SDK is placed in the SDK folder, which is located inside the OS Design workspace.

3 Configuring Microsoft eMbedded Visual C++ for PVR and OGLES Tools

To configure Visual C++, install the SDK as mentioned in Section 2, "Creating New SDK Containing MBX and VPU Functionality in WinCE Platform Builder." If the SDK is installed correctly, the name of SDK is displayed on the top of the screen and programs for the i.MX31 PDK can be compiled.

Figure 6 shows an eMbedded Visual C++ window.



test | 3DSMobility | Win32 (WCE ARMV4I) Debug | 3DSMobility Device

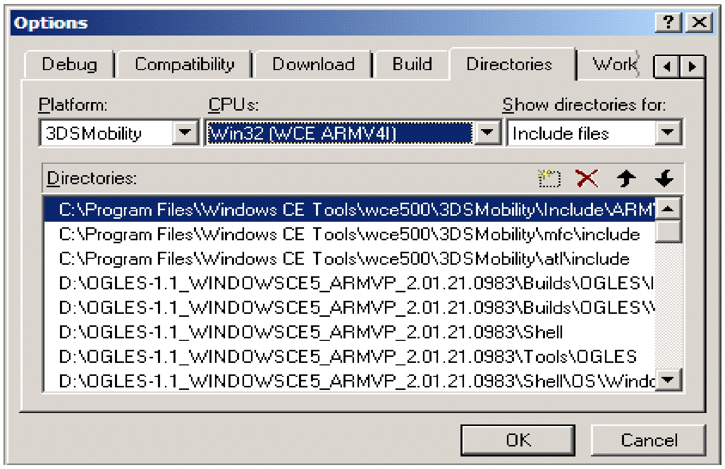
Workspace 'test': 1 project(s)
test files

Figure 6. Microsoft eMbedded Visual C++ Window

To use the PVR SDK for building 3D applications, OpenGL 1.1 SDK has to be downloaded for the MBX family. Go to http://www.imgtec.com/powervr/insider/powervr-sdk.asp to download OpenGL 1.1 SDK. To compile using PVR SDK, a series of paths have to be given to the linker. The paths can be given only once, while setting up the paths required by the linker during the platform configuration (the SDK is built in the platform builder).

Steps to configure the eMbedded Visual C++ for PVR and OGLES tools are as follows:

1. Go to Tools > Options > Directories in the Microsoft eMbedded Visual C++ window.
2. Select the platform as Installed SDK in the Platform Builder, and choose the other settings as shown in Figure 7.



Platform: 3DSMobility

CPUs: Win32 (WCE ARMV4I)

Show directories for: Include files

Directories:

```
C:\Program Files\Windows CE Tools\wce500\3DSMobility\Include\ARM
C:\Program Files\Windows CE Tools\wce500\3DSMobility\mfc\include
C:\Program Files\Windows CE Tools\wce500\3DSMobility\atl\include
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Builds\OGLES\I
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Builds\OGLES\\
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Shell
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Tools\OGLES
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Shell\OS\Windc
```

Figure 7. Options Window with Show Directories for: Include Files

Add the following paths, and click OK to close the dialog box:

— <PVR SDK Path>\Builds\OGLES\Include

— <PVR SDK Path>\Builds\OGLES\Windows\Include

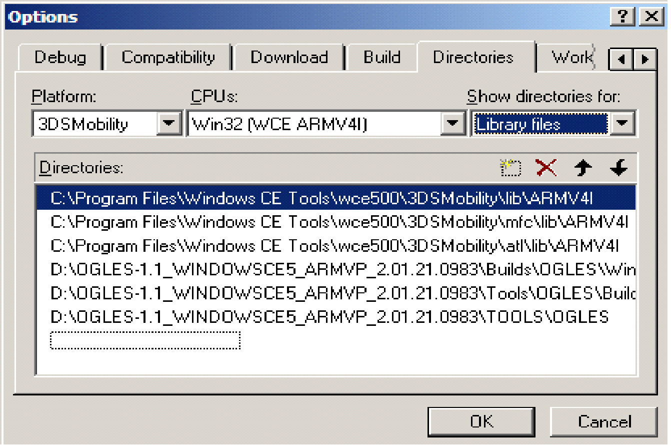
— <PVR SDK Path>\Shell

— <PVR SDK Path>\Tools\OGLES

— <PVR SDK Path>\Shell\OS\Windows

— <PVR SDK Path>\Shell\API\KEGL

3. Select the platform as Installed SDK in the Platform Builder, and use the other settings as shown in Figure 8.



Platform: 3DSMobility

CPUs: Win32 (WCE ARMV4I)

Show directories for: Library files

Directories:

```
C:\Program Files\Windows CE Tools\wce500\3DSMobility\lib\ARMV4I
C:\Program Files\Windows CE Tools\wce500\3DSMobility\mfc\lib\ARMV4I
C:\Program Files\Windows CE Tools\wce500\3DSMobility\atl\lib\ARMV4I
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Builds\OGLES\Win
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Tools\OGLES\Build
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\TOOLS\OGLES
```

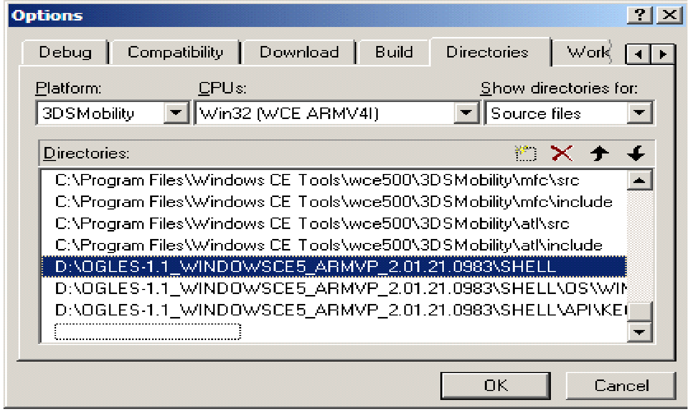
Figure 8. Options Window with Show Directories for: Library Files

Add the following paths, and click OK to close the dialog box:

— <PVR SDK Path> \Builds\OGLES\WindowsCE5ARMV4I\lib

— <PVR SDK Path> \Tools\OGLES\Build\WindowsCE\Common-ARMV4IRel

4. Select the platform as Installed SDK in the Platform Builder, and use the other settings as shown in Figure 9.



Platform: 3DSMobility

CPUs: Win32 (WCE ARMV4I)

Show directories for: Source files

Directories:

```
C:\Program Files\Windows CE Tools\wce500\3DSMobility\mfc\src
C:\Program Files\Windows CE Tools\wce500\3DSMobility\mfc\include
C:\Program Files\Windows CE Tools\wce500\3DSMobility\atl\src
C:\Program Files\Windows CE Tools\wce500\3DSMobility\atl\include
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\SHELL
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\SHELL\OS\WIN
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\SHELL\API\KE
```

Figure 9. Options Window with Show Directories for: Source Files

Add the following paths, and click OK to close the dialog box:

— <PVR SDK Path>\SHELL

— <PVR SDK Path>\SHELL\OS\WINDOWS

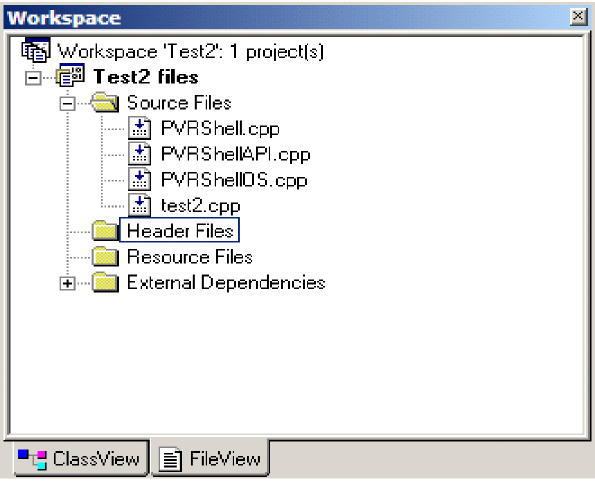
— <PVR SDK Path>\SHELL\API\KEGL

NOTE

Compilation in the selected platform causes the linker to access the PVR SDK paths.

When a new project is created, the following operations have to be done:

1. To create a new project, PVR SDK has to add the source files: PVRShell.cpp, PVRShellAPI.cpp, and PVRShellOS.cpp. The FileView of the project should look as shown in Figure 10.



Workspace

```
Workspace 'Test2': 1 project(s)
  Test2 files
    Source Files
      PVRShell.cpp
      PVRShellAPI.cpp
      PVRShellOS.cpp
      test2.cpp
    Header Files
    Resource Files
    External Dependencies
```

Figure 10. FileView

NOTE

The main source file must inherit a class from PVRShell and follow the PVRShell template. The template can be found in the path <PVR SDK Path>\TrainingCourse\03_IntroducingPVRShell.

2. The following lines should be added in the source code to make the compiler to link the PVR libraries:

— #pragma comment(lib, "libGLES_CM.lib")

— #pragma comment(lib, "OGLESTools.lib")

4 Configuring Microsoft Visual Studio® 2005 for PVR and OGLES Tools

Steps to configure Visual Studio 2005 are as follows:

1. Go to Tools > Options, as shown in Figure 11.

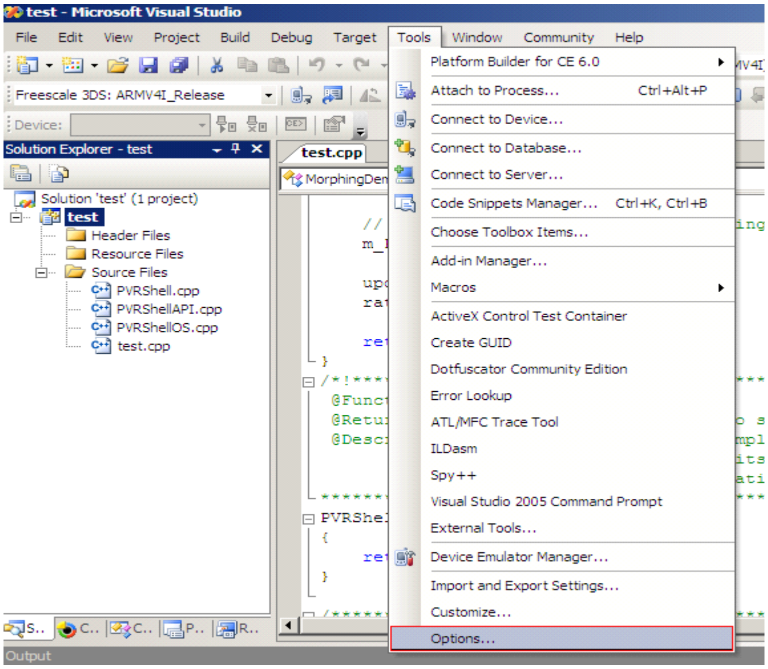
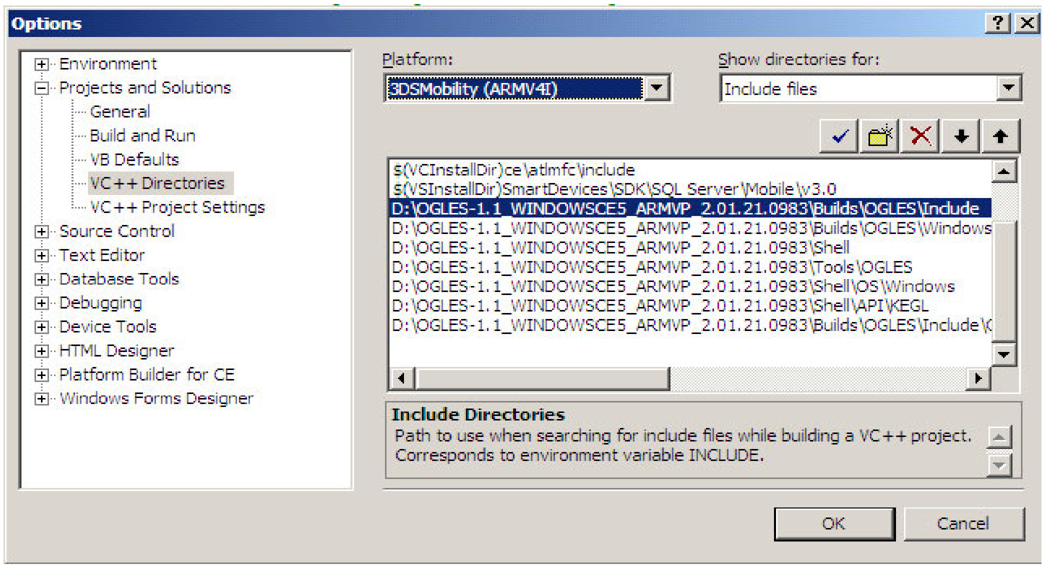


Figure 11. Microsoft Visual Studio Window

2. Go to Projects and Solutions > VC++ Directories in the Options menu, and select the settings as shown in Figure 12.



Platform: 3DSMobility (ARMV4I)

Show directories for: Include files

```
$(VCInstallDir)ce\atlmfc\include
$(VSInstallDir)SmartDevices\SDK\SQL Server\Mobile\v3.0
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Builds\OGLES\Include
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Builds\OGLES\Windows
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Shell
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Tools\OGLES
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Shell\OS\Windows
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Shell\API\KEGL
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Builds\OGLES\Include\(
```

Include Directories
Path to use when searching for include files while building a VC++ project. Corresponds to environment variable INCLUDE.

Figure 12. Options Menu with Show Directories for: Include File

Add the following paths, and click OK to close the window:

— <PVR SDK Path>\Builds\OGLES\Include

— <PVR SDK Path>\Builds\OGLES\Windows\Include

— <PVR SDK Path>\Shell

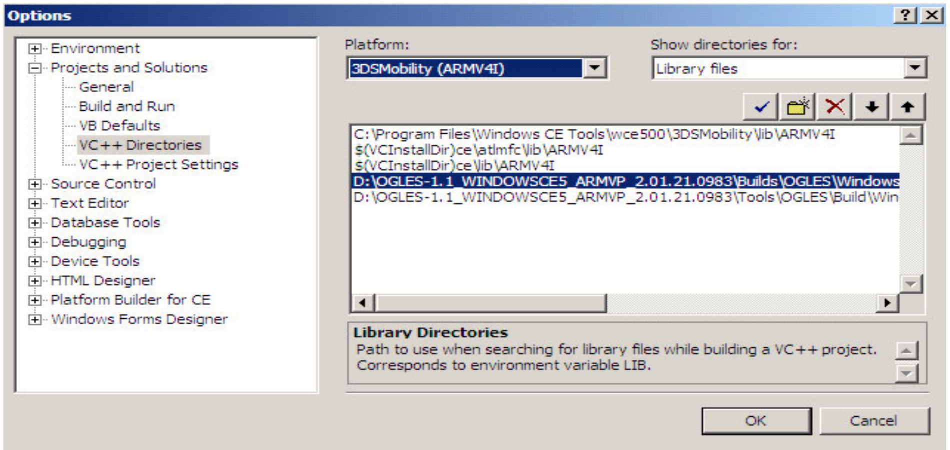
— <PVR SDK Path>\Tools\OGLES

— <PVR SDK Path>\Shell\OS\Windows

— <PVR SDK Path>\Shell\API\KEGL

— <PVR SDK Path>\Builds\OGLES\Include\GLES

3. In the Options menu, select the settings as shown in Figure 13.



Platform: 3DSMobility (ARMV4I)

Show directories for: Library files

```
C:\Program Files\Windows CE Tools\wce500\3DSMobility\lib\ARMV4I
$(VCInstallDir)ce\atlmfc\lib\ARMV4I
$(VCInstallDir)ce\lib\ARMV4I
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Builds\OGLES\Windows
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Tools\OGLES\Build\Win
```

Library Directories
Path to use when searching for library files while building a VC++ project. Corresponds to environment variable LIB.

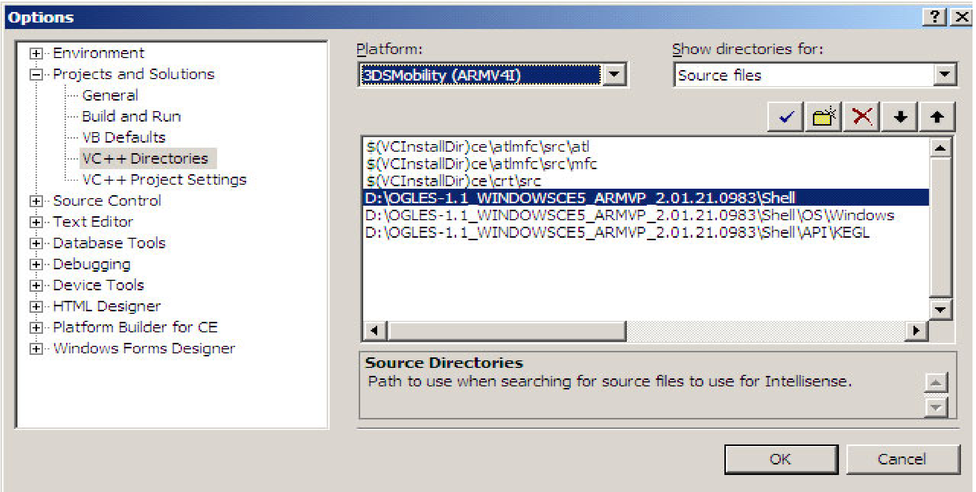
Figure 13. Options Menu with Show Directories for: Library Files

Add the following paths, and click OK to close the window:

— <PVR SDK Path> \Builds\OGLES\WindowsCE5ARMV4I\lib

— <PVR SDK Path> \Tools\OGLES\Build\WindowsCE\Common-ARMV4IRel

4. In the Options menu, select the settings as shown in Figure 14.



Platform: 3DSMobility (ARMV4I)

Show directories for: Source files

```
$(VCInstallDir)ce\atlmfc\src\atl
$(VCInstallDir)ce\atlmfc\src\mfc
$(VCInstallDir)ce\crt\src
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Shell
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Shell\OS\Windows
D:\OGLES-1.1_WINDOWSCE5_ARMVP_2.01.21.0983\Shell\API\KEGL
```

Source Directories
Path to use when searching for source files to use for Intellisense.

Figure 14. Options Menu with Show Directories for: Source Files

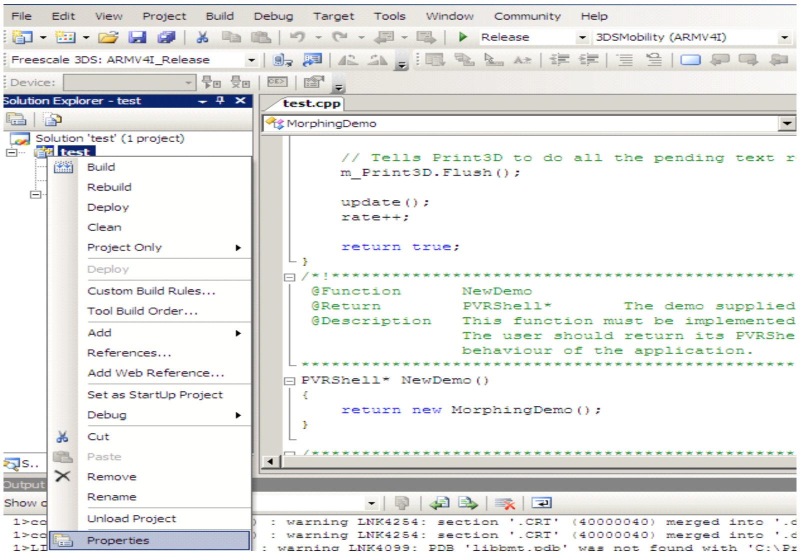
Add the following paths, and click OK to close the window:

— <PVR SDK Path>\SHELL

— <PVR SDK Path>\SHELL\OS\WINDOWS

— <PVR SDK Path>\SHELL\API\KEGL

5. In the Solution Explorer, right-click on the Project and select Properties as shown in Figure 15.



```
    // Tells Print3D to do all the pending text r
    m_Print3D.Flush();

    update();
    rate++;

    return true;
}
/*!*********************************************
 @Function       NewDemo
 @Return         PVRShell*       The demo supplied
 @Description    This function must be implemented
                 The user should return its PVRShe
                 behaviour of the application.
***********************************************
PVRShell* NewDemo()
{
    return new MorphingDemo();
}

/**********************************************
```

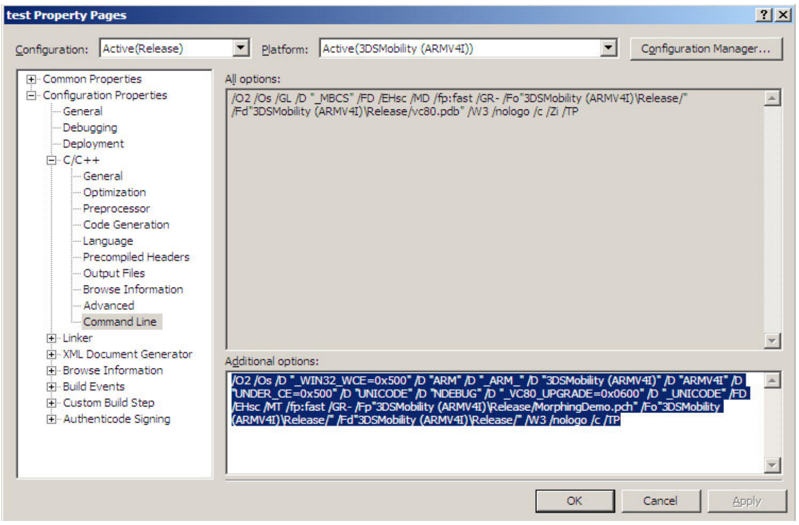
```
) : warning LNK4254: section '.CRT' (40000040) merged into '.c
) : warning LNK4254: section '.CRT' (40000040) merged into '.c
: warning LNK4099: PDB 'libbmt.pdb' was not found with 'C:\Pr
```

Figure 15. Solution Explorer

6. Go to Configuration Properties > C/C++ > Command Line in the Property Page, and add the following additional options:

```
/O2 /Os /D "_WIN32_WCE=0x500" /D "ARM" /D "_ARM_" /D "3DSMobility (ARMV4I)" /D
"ARMV4I" /D
"UNDER_CE=0x500" /D "UNICODE" /D "NDEBUG" /D "_VC80_UPGRADE=0x0600" /D "_UNICODE" /FD
/EHsc /MT /fp:fast /GR- /Fp"3DSMobility (ARMV4I)\Release/MorphingDemo.pch"
/Fo"3DSMobility
(ARMV4I)\Release/" /Fd"3DSMobility (ARMV4I)\Release/" /W3 /nologo /c /TP
```

The Property Page after the addition should look as shown in Figure 16.



All options:

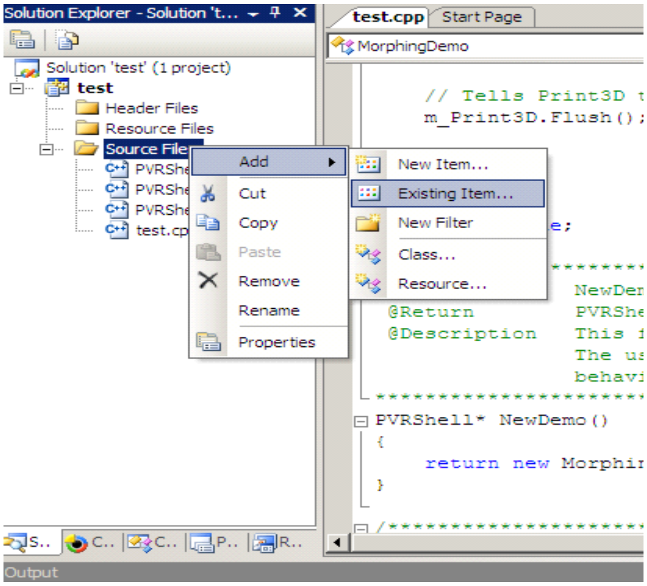
```
/O2 /Os /GL /D "_MBCS" /FD /EHsc /MD /fp:fast /GR- /Fo"3DSMobility (ARMV4I)\Release/"
/Fd"3DSMobility (ARMV4I)\Release/vc80.pdb" /W3 /nologo /c /Zi /TP
```

Additional options:

```
/O2 /Os /D "_WIN32_WCE=0x500" /D "ARM" /D "_ARM_" /D "3DSMobility (ARMV4I)" /D "ARMV4I" /D
"UNDER_CE=0x500" /D "UNICODE" /D "NDEBUG" /D "_VC80_UPGRADE=0x0600" /D "_UNICODE" /FD
/EHsc /MT /fp:fast /GR- /Fp"3DSMobility (ARMV4I)\Release/MorphingDemo.pch" /Fo"3DSMobility
(ARMV4I)\Release/" /Fd"3DSMobility (ARMV4I)\Release/" /W3 /nologo /c /TP
```

Figure 16. Property Page

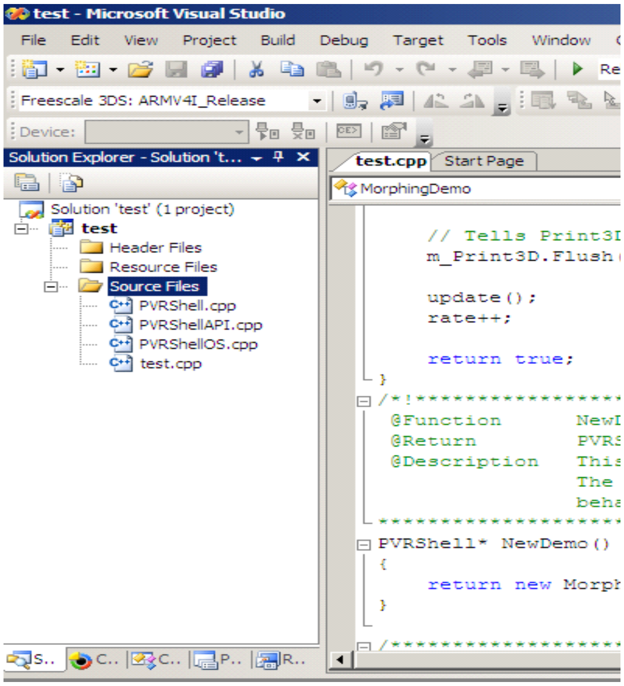
7. Right-click on the Source files in the Solution Explorer, and select Add > Existing Item, as shown in Figure 17.



```
    // Tells Print3D t
    m_Print3D.Flush();

e;

**********
                NewDem
@Return         PVRShe
@Description    This i
                The us
                behavi
**********************
PVRShell* NewDemo()
{
    return new Morphi
}

/**********************
```

Figure 17. Solution Explorer Before Adding the Files

8. Add the files: PVRShell.cpp, PVRShellAPI.cpp, and PVRShellOS.cpp in the PVR SDK Package, and the Solution Explorer window should look as shown in Figure 18.



```
    // Tells Print3I
    m_Print3D.Flush(

    update();
    rate++;

    return true;
}
/*!*******************
 @Function      NewI
 @Return        PVRS
 @Description   This
                The
                beha
*********************
PVRShell* NewDemo()
{
    return new Morph
}
/********************
```

Figure 18. Solution Explorer After Adding the Files

NOTE

The new file should have a main function, so that it can compile. To use the PVRShell, the program must inherit from PVRShell—class Test: public PVRShell.

5 References

The reference for the application note is given as follows:

- *MCIMX31 and MCIMX31L Applications Processors Reference Manual* (MCIMX31RM)

6 Revision History

Table 1 provides a revision history for this application note.

Table 1. Document Revision History

Rev. Number	Date	Substantive Change(s)
0	05/2010	Initial release.

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products 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 Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor 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 Freescale Semiconductor was negligent regarding the design or manufacture of the part.