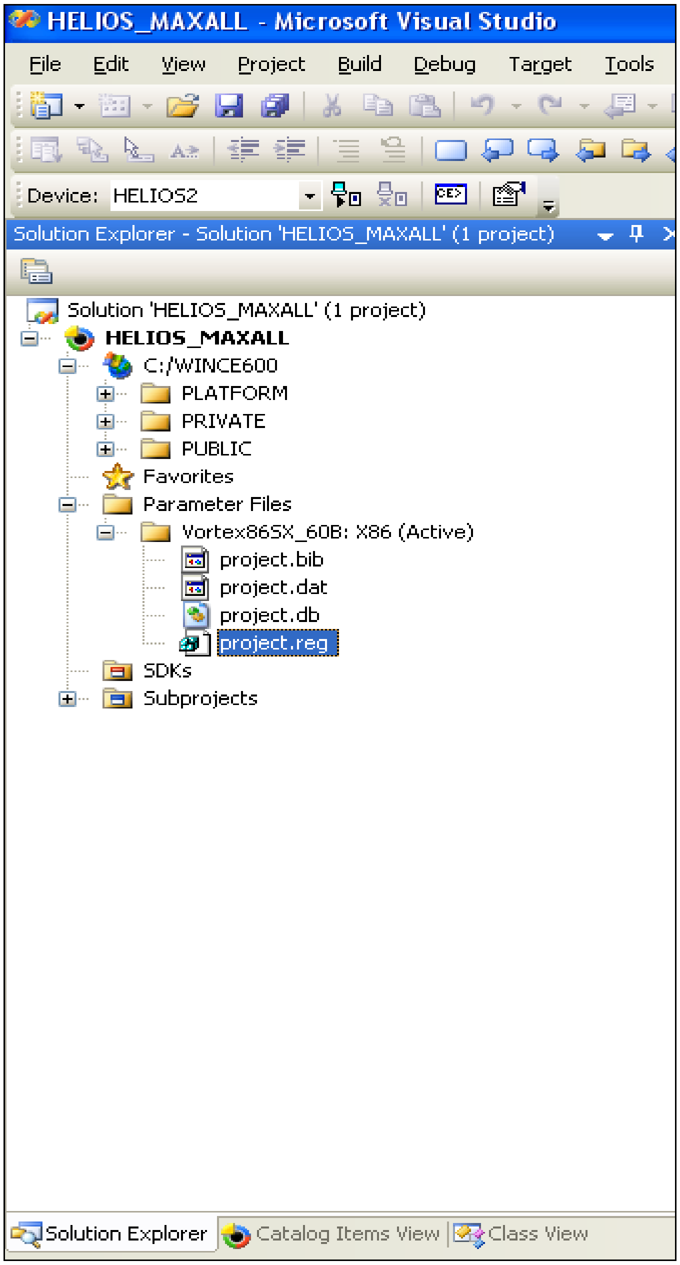
Developing Applications using Universal Driver 6.0x in Windows CE 6.0

Rev. B

1 Installing Universal Driver 6.0x in the platform

The installation of UNIVERSAL DRIVER is a very simple process and requires the DSCUD_API.DLL and DSCUD.DLL files be present in the release directories of the platform. DSCUD.DLL is required to be present in the project.reg registry file. The Project.reg file can be found in the Solutions explorer tab of Platform Builder as shown in the picture below.


HELIOS_MAXALL - Microsoft Visual Studio
File Edit View Project Build Debug Target Tools
Device: HELIOS2
Solution Explorer - Solution 'HELIOS_MAXALL' (1 project)
Solution 'HELIOS_MAXALL' (1 project)
HELIOS_MAXALL
C:/WINCE600
PLATFORM
PRIVATE
PUBLIC
Favorites
Parameter Files
Vortex86SX_60B: X86 (Active)
project.bib
project.dat
project.db
project.reg
SDKs
Subprojects
Solution Explorer
Catalog Items View
Class View

The project.reg file must have the following entries for UNIVERSAL DRIVER to work.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\DSCUD]
      "Dll"="dscud.dll"
      "Prefix"="DSC"
      "Order"=dword:1
      "SysIntr"=dword:15
      "Index"=dword:1
      "Flags"=dword:10
      "UserProcGroup"=dword:3
```

The above entries are required so that the kernel can load DSCUD.DLL file automatically when the operating system boots. All the above described entries must be present in the registry. The SysIntr value is the value of the IRQ used by the DAQ circuitry + SYSINTR_FIRMWARE (0x10). If the board is going to use some other IRQ, the SysIntr value in the registry MUST be changed to reflect the change.

If the SysIntr value does not correspond to the IRQ for the DAQ circuit, the driver will not be able to handle interrupt requests.

2 Developing application using Universal Driver on Windows CE 6.0.

To develop an application using UD 6.0x on Windows CE 6.0, all you need is the Visual Studio 2005 with Platform Builder 6.0 installed on your system.

The Vortex 86 SDK also must be installed in your system to develop applications using the SDK.

To start the development, any of the example programs provided by DSC can be used. The example programs are all C programs and are simple console applications which can be used on Windows CE 6.0 after recompiling.

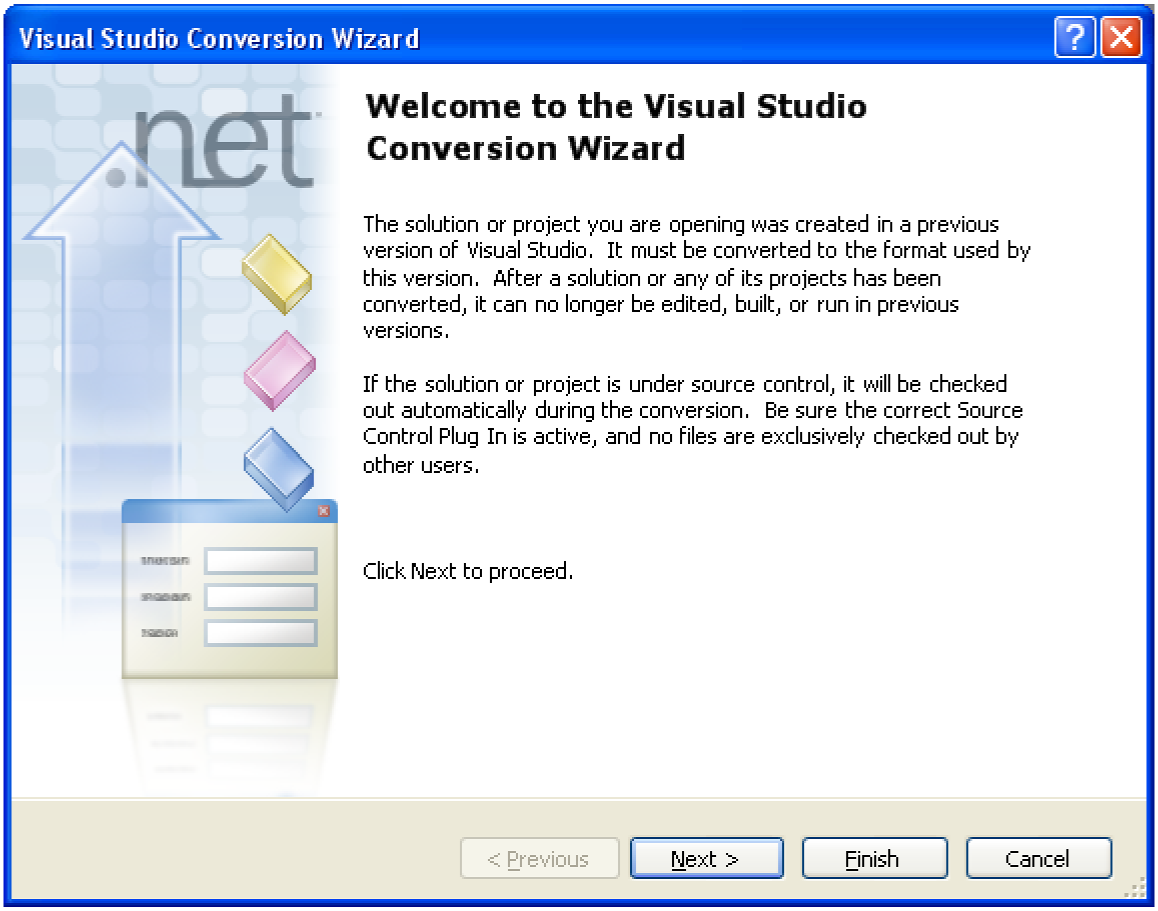
For all the example programs, the UD header and library files are required. The UD header file MUST be included in the example program and the UD library file (DSCDU_API.LIB) must be linked to by the example program.

Header File	: DSCUD.H
Library File	: DSCUD_API.LIB
DLLs	: DSCUD_API.DLL (loaded when the example program launches) DSCUD.DLL (loaded automatically by the kernel on OS bootup)

Using an Existing Example Program:
All the example programs have a WinCE directory. From the Helios folder, select the application (DSCADSample) and browse to the WinCE directory.
Open the DSCADSample.VCW or DSCADSample.VCP file into the Visual Studio.

For the projects with the older extensions to be used on Visual Studio 2005, a conversion wizard will pop-up. Please upgrade the project files as instructed by the wizard.

When a project file is opened, the following window will open up…



Welcome to the Visual Studio Conversion Wizard

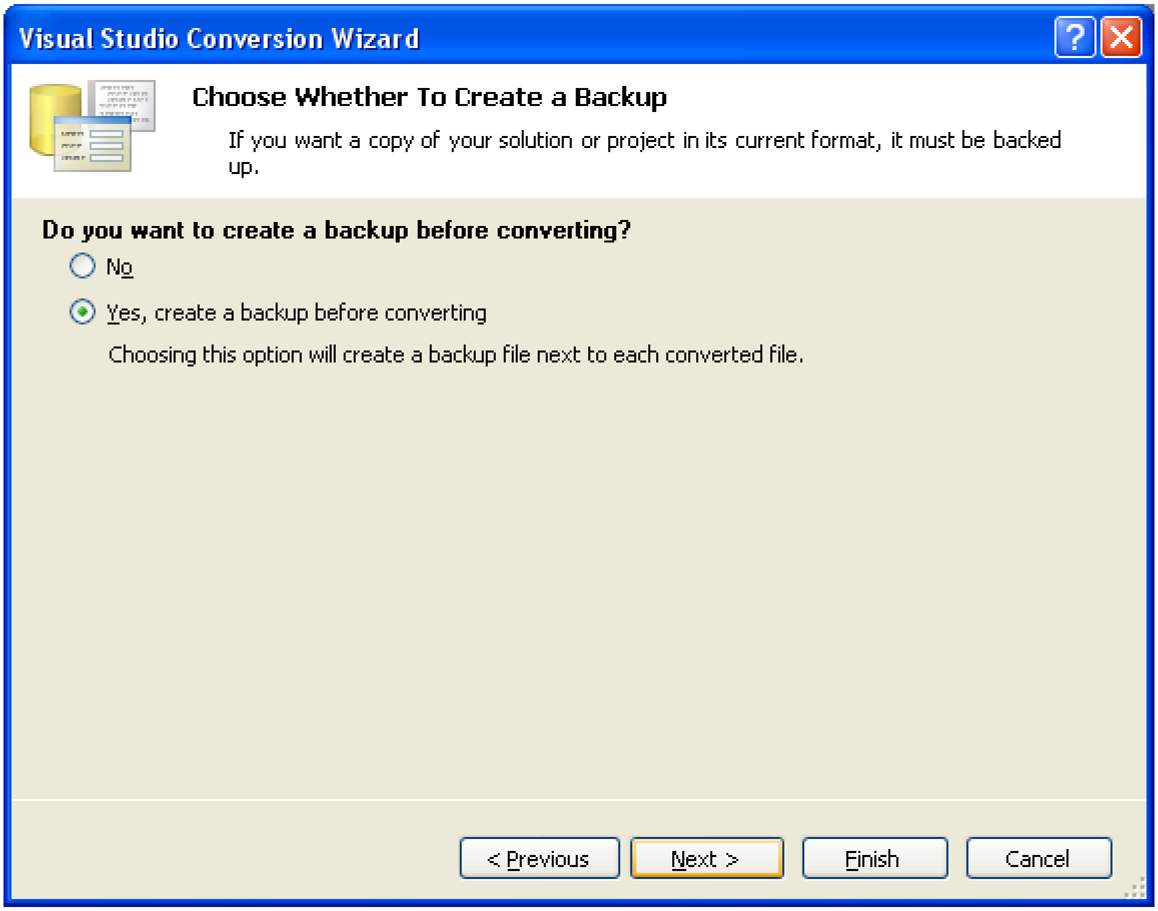
The solution or project you are opening was created in a previous version of Visual Studio. It must be converted to the format used by this version. After a solution or any of its projects has been converted, it can no longer be edited, built, or run in previous versions.

If the solution or project is under source control, it will be checked out automatically during the conversion. Be sure the correct Source Control Plug In is active, and no files are exclusively checked out by other users.

Click Next to proceed.

Click Next.

On the next window, select Yes for backing up the original file and click Next. Even No can be selected on this screen.



Choose Whether To Create a Backup

If you want a copy of your solution or project in its current format, it must be backed up.

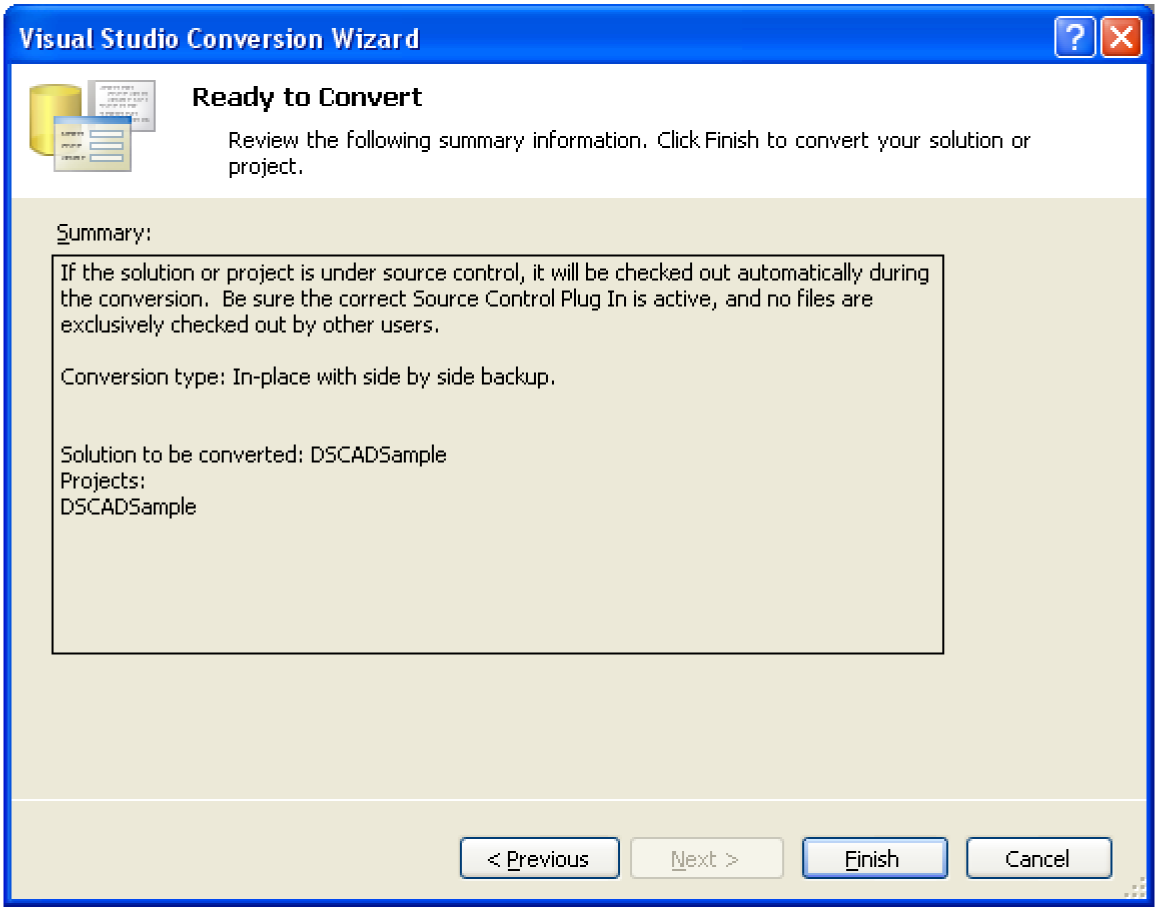
Do you want to create a backup before converting?

○ No

● Yes, create a backup before converting

Choosing this option will create a backup file next to each converted file.

The final window will be as below…



Ready to Convert

Review the following summary information. Click Finish to convert your solution or project.

Summary:

If the solution or project is under source control, it will be checked out automatically during the conversion. Be sure the correct Source Control Plug In is active, and no files are exclusively checked out by other users.

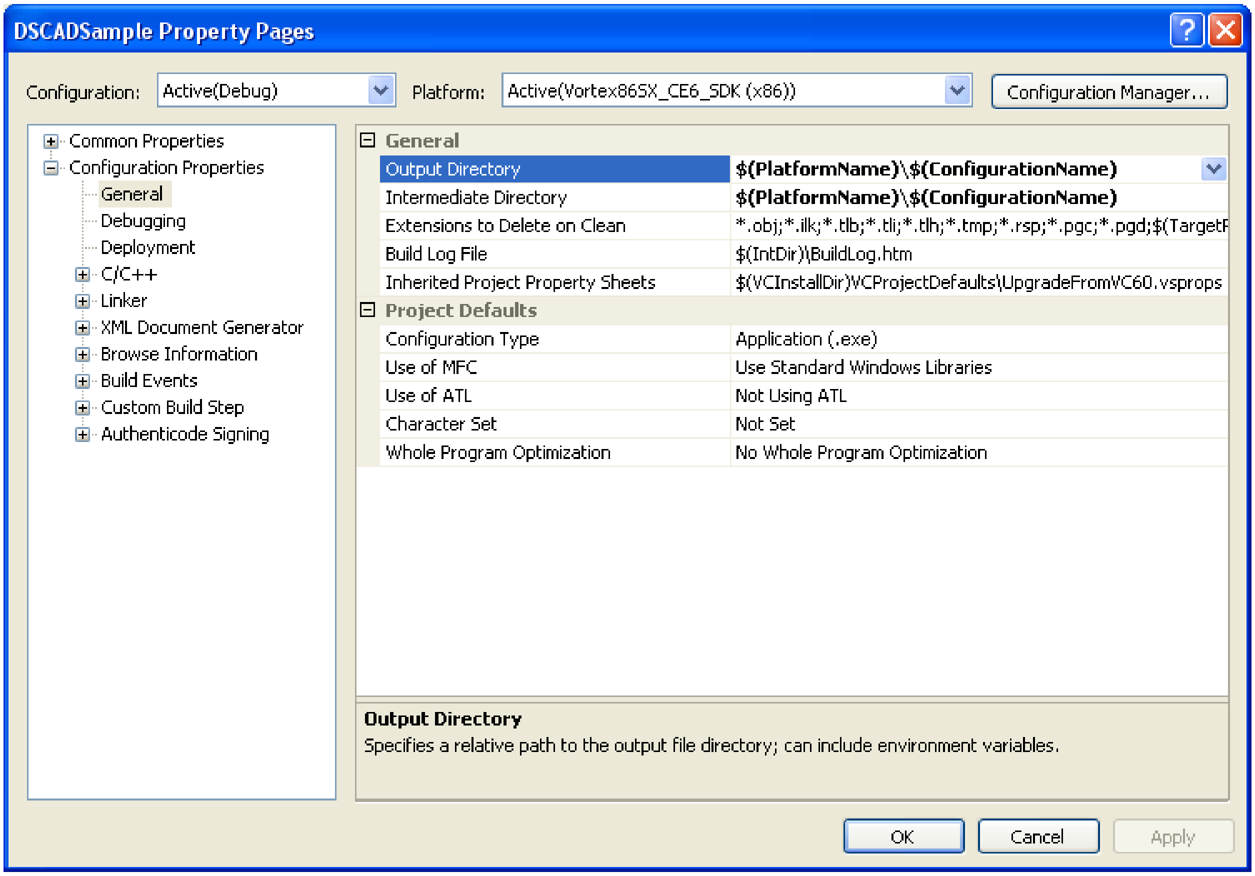
Conversion type: In-place with side by side backup.

Solution to be converted: DSCADSample
Projects:
DSCADSample

Click Finish.

The files will be saved as DSCADSample.vcproj and another file will be created with an extension of .SLN. In future, these files should be used for the usage.

From the Solutions Explorer window, right click on DSCADSample and select Properties. The following window will open up.



DSCADSample Property Pages

Configuration: Active(Debug)

Platform: Active(Vortex86SX_CE6_SDK (x86))

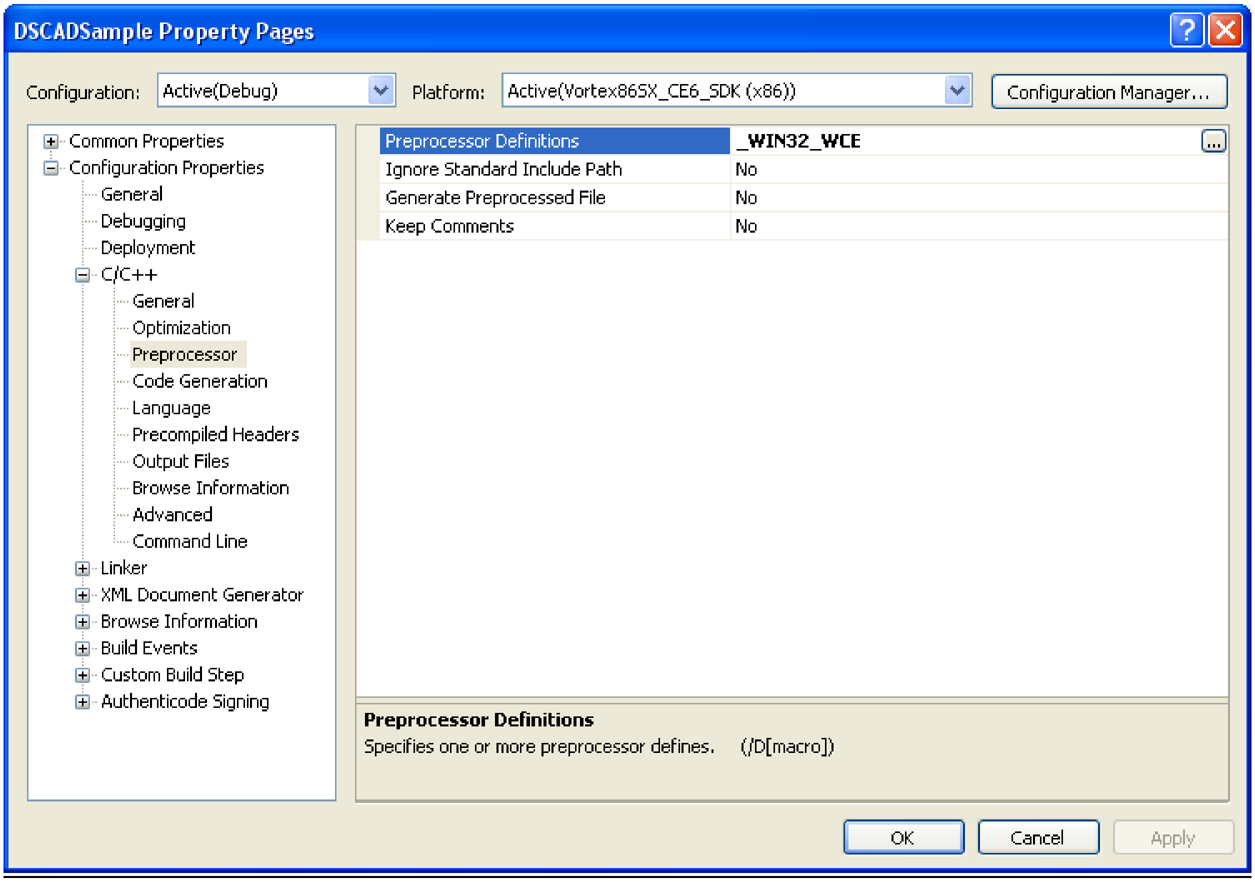
General	
Output Directory	**$(PlatformName)\$(ConfigurationName)**
Intermediate Directory	**$(PlatformName)\$(ConfigurationName)**
Extensions to Delete on Clean	*.obj;*.ilk;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgd;$(TargetF
Build Log File	$(IntDir)\BuildLog.htm
Inherited Project Property Sheets	$(VCInstallDir)VCProjectDefaults\UpgradeFromVC60.vsprops
Project Defaults	
Configuration Type	Application (.exe)
Use of MFC	Use Standard Windows Libraries
Use of ATL	Not Using ATL
Character Set	Not Set
Whole Program Optimization	No Whole Program Optimization

Output Directory

Specifies a relative path to the output file directory; can include environment variables.

PREPROCESSOR DEFINITION:

Enter _WIN32_WCE in the preprocessor definitions text box as shown below.



DSCADSample Property Pages

Configuration: Active(Debug)

Platform: Active(Vortex86SX_CE6_SDK (x86))

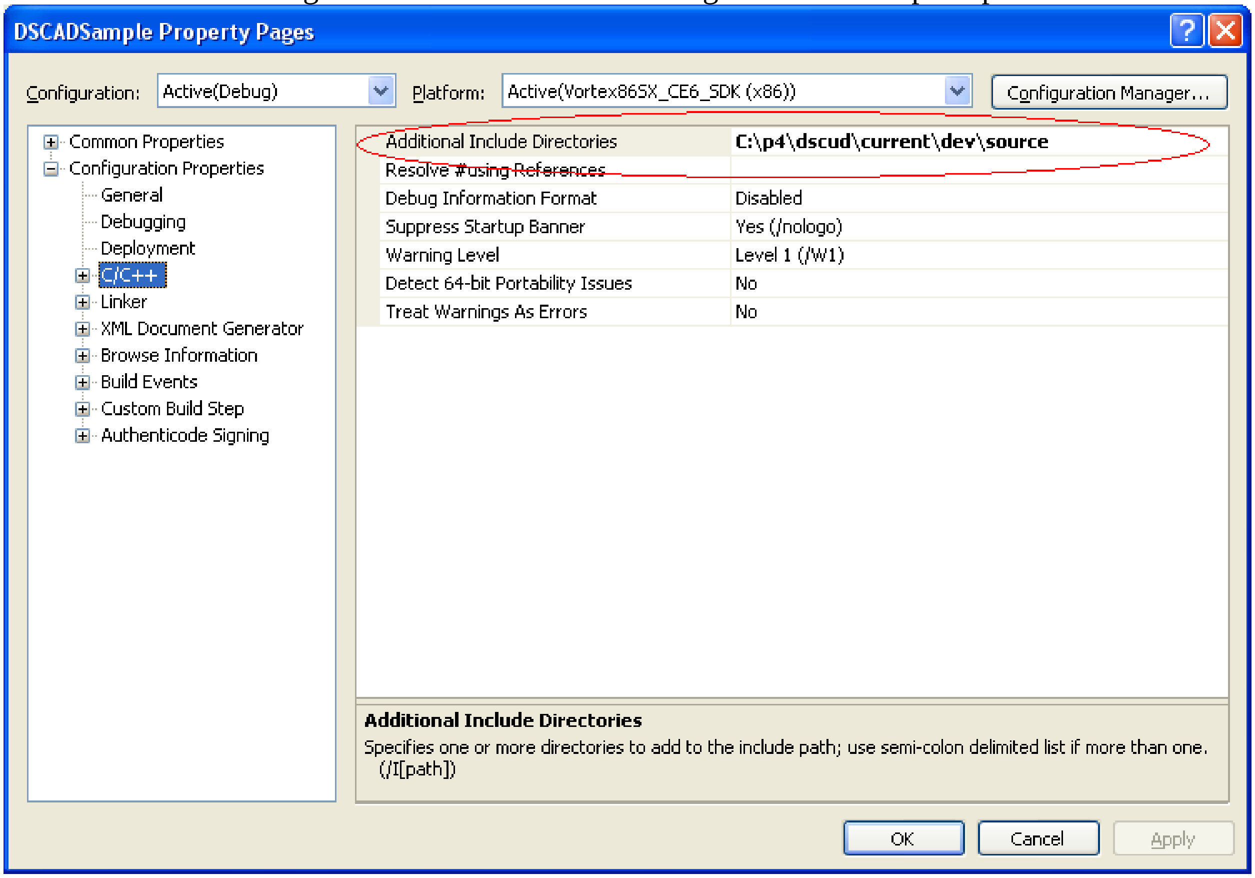
Preprocessor Definitions	_WIN32_WCE
Ignore Standard Include Path	No
Generate Preprocessed File	No
Keep Comments	No

Preprocessor Definitions

Specifies one or more preprocessor defines. (/D[macro])

INCLUDE DIRECTORIES:

The include file can be set to the folder where the required files are present. To set this, select the C/C++ link in the configuration window. The following window will open up.



Configuration: Active(Debug)

Platform: Active(Vortex86SX_CE6_SDK (x86))

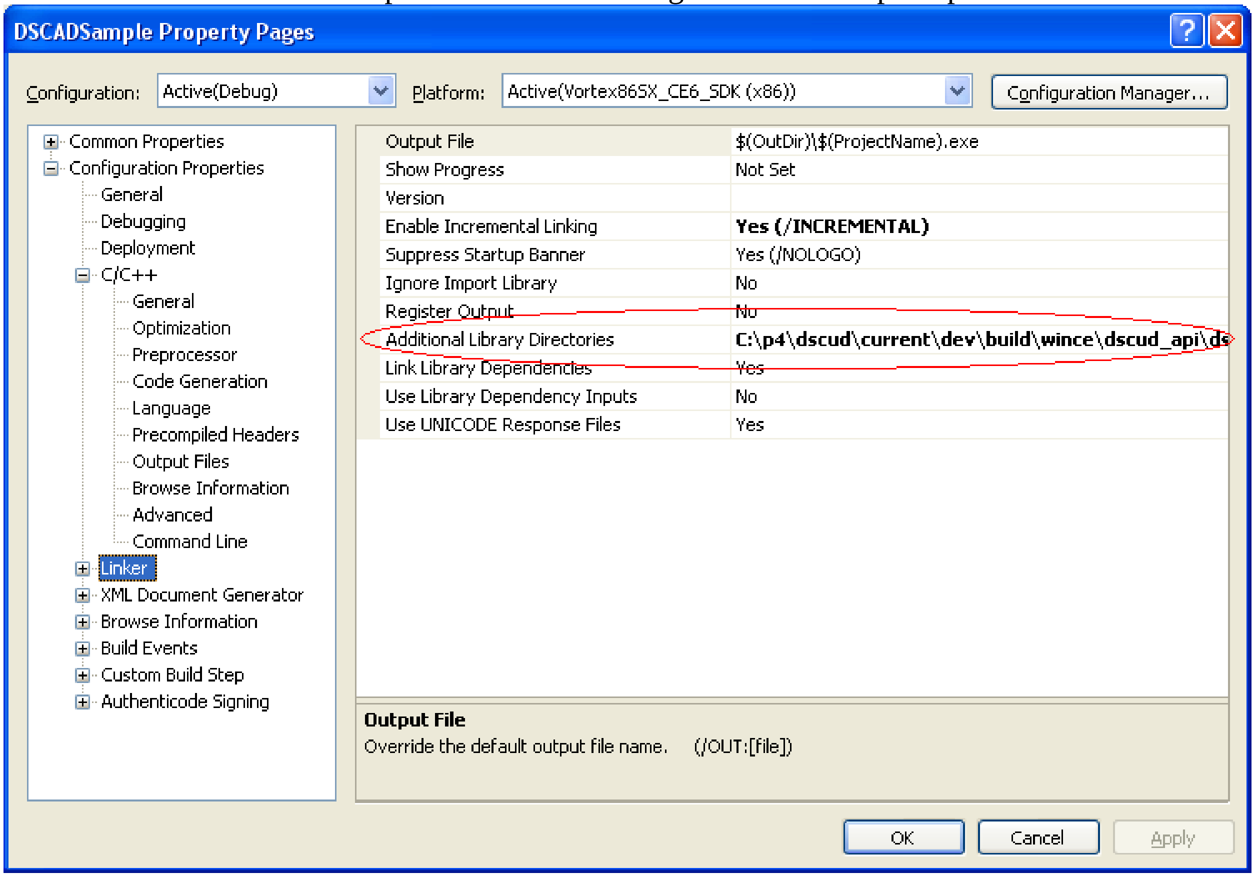
Additional Include Directories	C:\p4\dscud\current\dev\source
Resolve #using References	
Debug Information Format	Disabled
Suppress Startup Banner	Yes (/nologo)
Warning Level	Level 1 (/W1)
Detect 64-bit Portability Issues	No
Treat Warnings As Errors	No

Additional Include Directories
Specifies one or more directories to add to the include path; use semi-colon delimited list if more than one. (/I[path])

The directory path can be set in the Additional Include Directories configuration.

LIBRARY SETTINGS:

To set the path for the linked library dscud_api.lib, go to the configurations window as described earlier and select the Linker option and the following window will open up.



Configuration: Active(Debug)

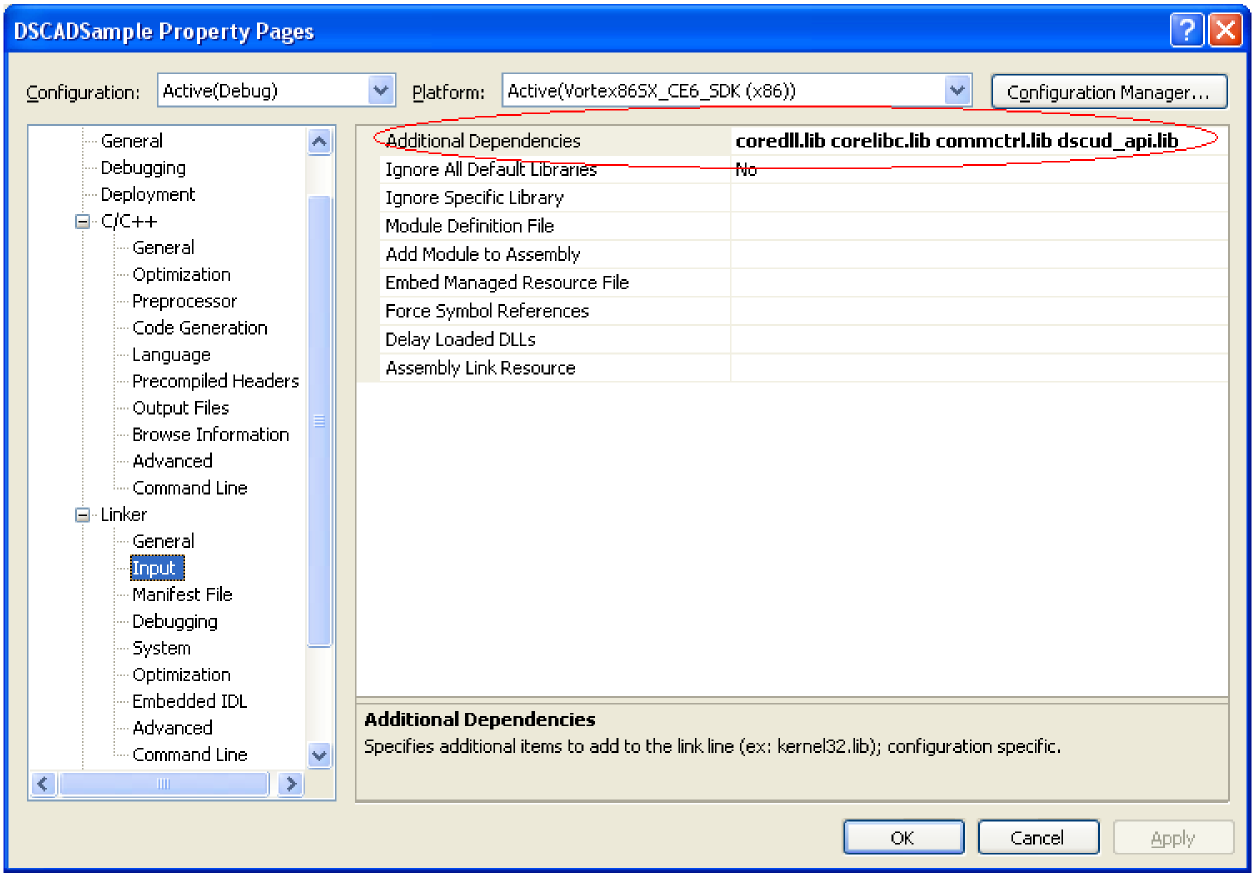
Platform: Active(Vortex86SX_CE6_SDK (x86))

Output File	$(OutDir)\$(ProjectName).exe
Show Progress	Not Set
Version	
Enable Incremental Linking	**Yes (/INCREMENTAL)**
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Additional Library Directories	**C:\p4\dscud\current\dev\build\wince\dscud_api\ds**
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Use UNICODE Response Files	Yes

Output File
Override the default output file name. (/OUT:[file])

Specify the path where the dscud_api.lib is copied in the Additional Library Directories text box.

To specify that the application will use the dscud_api.lib static library, expand the Linker link and select the Input link. The following window will open up.



Configuration: Active(Debug)

Platform: Active(Vortex86SX_CE6_SDK (x86))

Additional Dependencies	**coredll.lib corelibc.lib commctrl.lib dscud_api.lib**
Ignore All Default Libraries	No
Ignore Specific Library	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded DLLs	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link line (ex: kernel32.lib); configuration specific.

Enter dscud_api.lib in the Additional Dependencies text box in the configuration window.

NOTE: After every change that is made in the configuration window, the Apply button must be clicked for the changes to be saved. After all the changes are complete, click OK button to close the configuration window and build the application.

An EXE will be made with the Project Name specified.

2.1 *Adding user exe to the platform*

The executable created as described in the previous section, can be added to the platform by adding the entry for the same in project.bib file in the modules section.

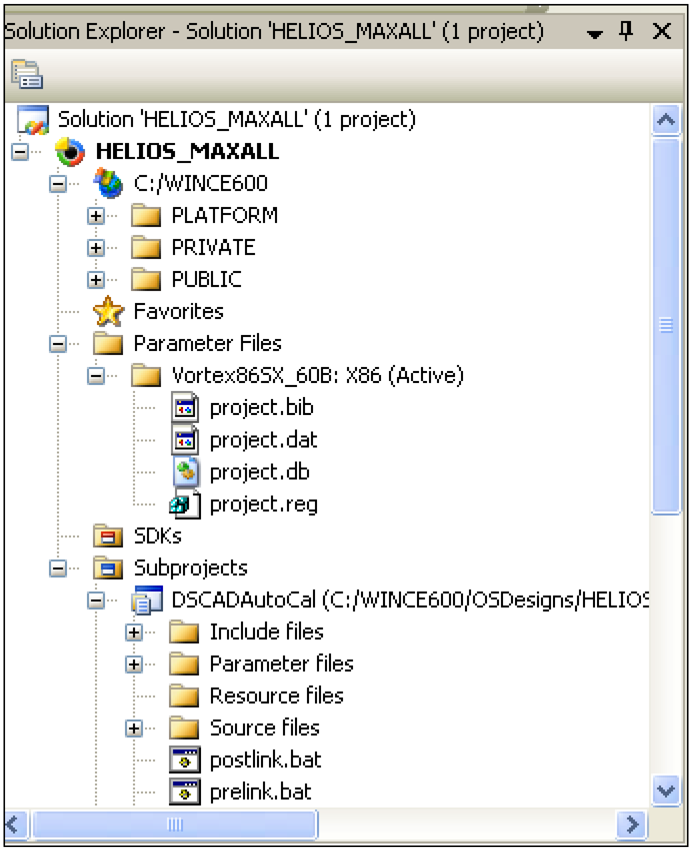
```
MODULES
;  Name            Path                                           Memory Type
;  --------------  ---------------------------------------------  -----------
  dscud.dll         $(_FLATRELEASEDIR)\dscud.dll                        NK  SH
  dscud_api.dll     $(_FLATRELEASEDIR)\dscud_api.dll                    NK  SH
  DSCADSample.exe   $(_FLATRELEASEDIR)\DSCADSample.exe                    NK
```

Make sure that the EXE is present in the platform's RelDir\Debug or RelDir\Release folder. If this entry is present in the project.BIB file, and the EXE is not present in the RelDir directories, the final stage of build process called makeimage will fail. If that happens, copy the EXE manually from the application's debug or release folder to the platform's RelDir\Debug or release folders and run the makeimage command again.

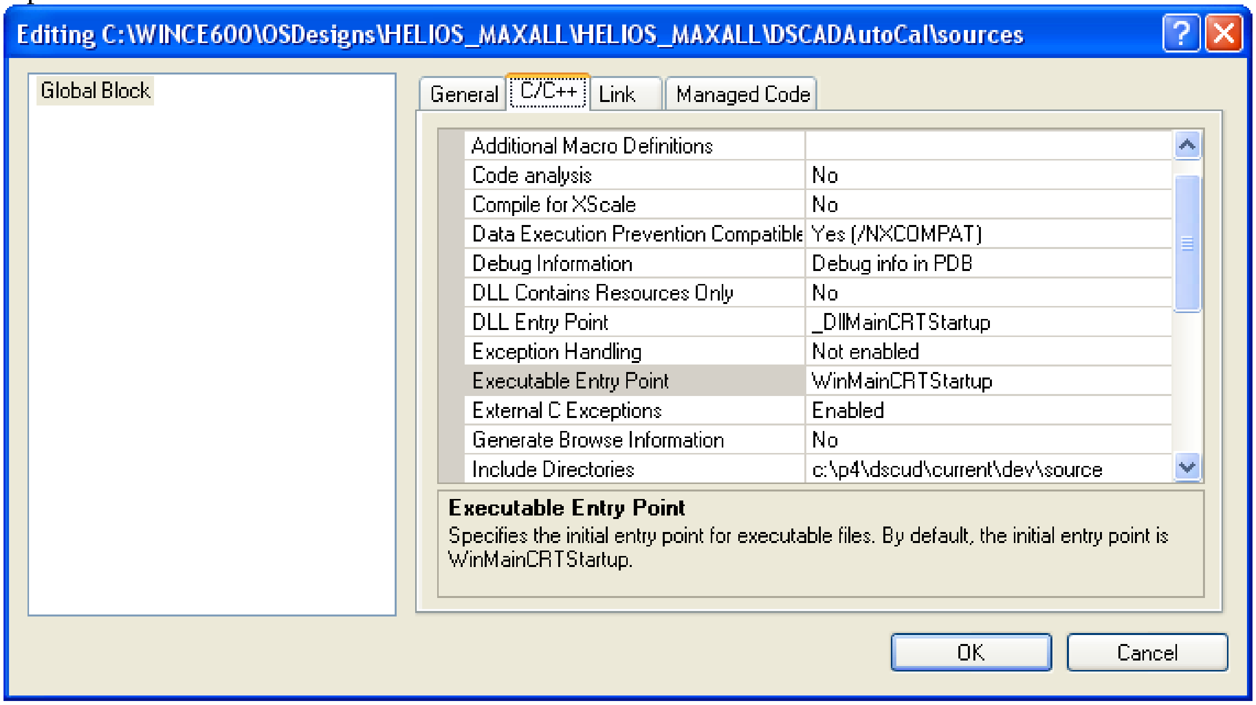
In order to just make the image, use the Build-Make Runtime image menu and the platform builder will build a new NK.BIN from the files already present in the Reldir folder(s).

2.2 *Making a subproject to the platform*

Another way of making a user program is to create a subproject in the platform builder. When a subproject option is used, the project is built along with the platform and thus is automatically included in the final NK.BIN.



Options for Include files…

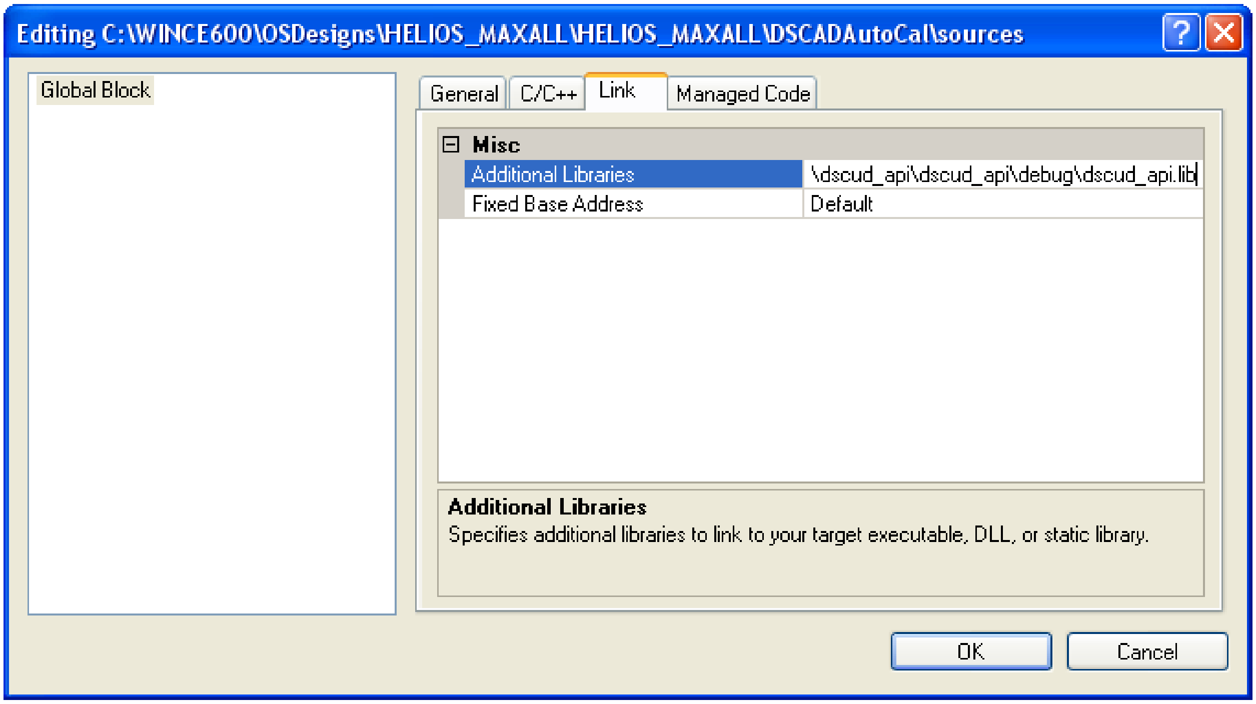


Editing C:\WINCE600\OSDesigns\HELIOS_MAXALL\HELIOS_MAXALL\DSCADAutoCal\sources

Additional Macro Definitions	
Code analysis	No
Compile for XScale	No
Data Execution Prevention Compatible	Yes (/NXCOMPAT)
Debug Information	Debug info in PDB
DLL Contains Resources Only	No
DLL Entry Point	_DllMainCRTStartup
Exception Handling	Not enabled
Executable Entry Point	WinMainCRTStartup
External C Exceptions	Enabled
Generate Browse Information	No
Include Directories	c:\p4\dscud\current\dev\source

Executable Entry Point
Specifies the initial entry point for executable files. By default, the initial entry point is WinMainCRTStartup.

Options for Linking lib files…



Editing C:\WINCE600\OSDesigns\HELIOS_MAXALL\HELIOS_MAXALL\DSCADAutoCal\sources

⊟ Misc	
Additional Libraries	\dscud_api\dscud_api\debug\dscud_api.lib
Fixed Base Address	Default

Additional Libraries
Specifies additional libraries to link to your target executable, DLL, or static library.

Again the files required to build a subproject application which uses UD are the include file DSCUD.H and DSCUD_API.LIB file for statically linking to the library.

The files required to be present in the RelDir\Debug or RelDir\Release directories are DSCUD_API.DLL and DSCUD.DLL.