

I have been using Compaq Visual Fortran 6.6 and Visual Studio 6.0 for years. I finally decided it is time to do whatever it takes to upgrade to the latest Fortran and C++ compilers. Neither of these old compilers are supported and they are not compatible with Windows Vista.

I have found that there is not that much information on how to efficiently upgrade to the new compilers. There are some simple examples of mixed Fortran programming that come with the Intel® compiler, but they were not quite complex enough to address the problems that I was encountering. However, you should refer to them because they do provide some insight into mixed language programming with the new compilers. Also, there is some documentation on mixed language programming and integrating with Visual Studio IDE. You should review all of this documentation if you can find it.

I am posting this in the hopes that you may learn from my experiences. I upgraded to Visual Studio 2008 and Compaq Visual Fortran 10.1. As a side note, I just needed to get these compilers working for my code and I have not investigated all aspects of either compiler and do not consider myself an expert.

My code consists of 10 of thousands of lines of C++ codes with comparable amount of Fortran code. The Fortran code is compiled as the primary project and the C code is compiled as a static object library in a C++ dependent project.

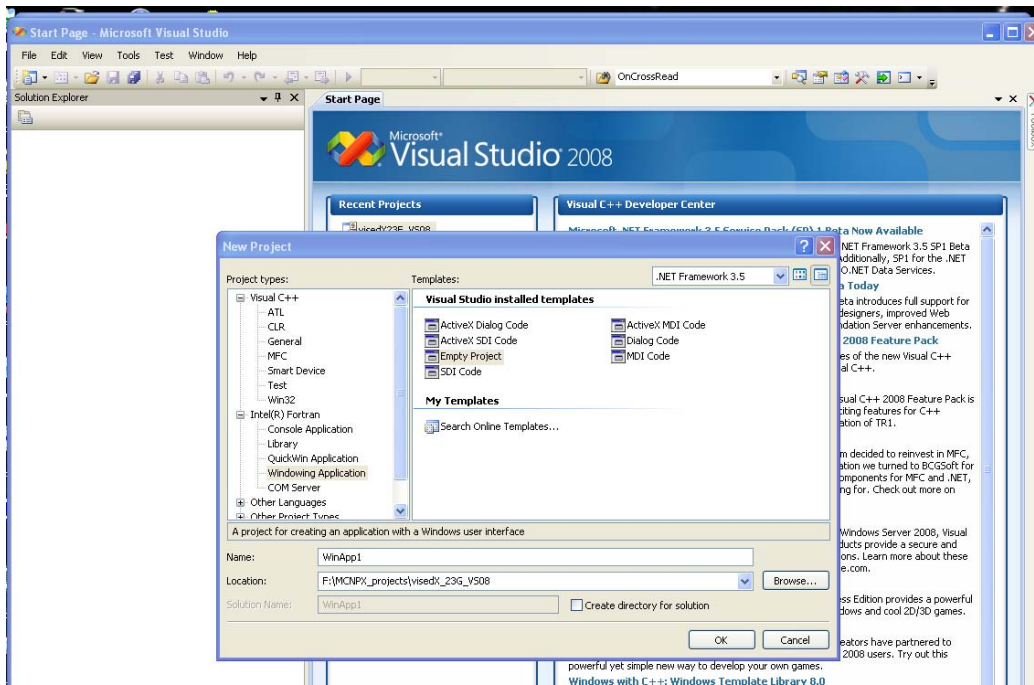
Under CVF/VS6 the code compiled in about 1 and a half minutes on my computer. After the upgrade to VS2008/INTEL® 10.1 it now compiles in about 5 and a half minutes. This is over three times longer.

Here are some installation notes about installing INTEL® Fortran with Visual Studio:
<http://www.intel.com/software/products/compilers/docs/fwin/install.htm>

Below of the key things to keep in mind when changing from CVF/VS6 to the new compilers:

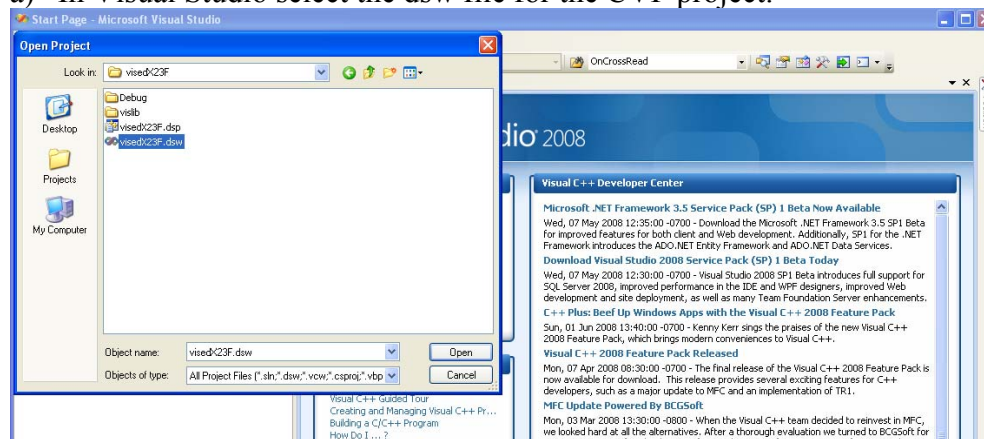
1. You need to install your Visual Studio software before installing the Fortran. I have both VS2005 and VS2008 installed on my system and Intel® Fortran works with both of them.
 - a) Install VS2008
 - b) Install any updates to VS2008
 - c) Install Intel® Fortran

To determine if you have installed the compilers correctly, in Visual Studio 2008, Select File->New Project and verify that Intel® Fortran is an option.



2. By far the easiest way to convert from CVF6.6/VS6 is to use the upgrade tools that are provided. I was hesitant to do this at first, but there are so many options that needed to be tweaked by hand that it takes a long time to determine the right combination and the upgrade tools take you most of the way. To do the automatic upgrade do the following:

a) In Visual Studio select the dsw file for the CVF project.



b) You will then be prompted to convert the project:



- c) Select “Yes”, since I am doing both a Fortran and C linked project , it will ask to convert the second dependent project.



- d) Select “Yes”, then select Project->Extract Compaq Visual Fortran Project Items.

- e) This will complete the conversion process. Now do a rebuild all and resolve any remaining issues that you encounter

3. Additional items that I had to address for my project are included below:

- a) In the C++ files replace
`#include <fstream.h>`

with

`#include <fstream>`
 using namespace std;

- b) In the C++ files replace
`#include <iostream.h>`

with

`#include <iostream>`
 using namespace std;

- c) If the VS6/CVF project had additional include directories set in VS6 (Tools->Options->Directories), you need to set them in the project.

- a. Right click on the Fortran project
- b. Select properties.

- c. Select the Fortran->General page
 - d. Set the include directory under “Additional Include Directories”
 - e. For my case:
F:\MCNPX_projects\MCNPX_SRC\MCNPX26E\v26e\src\include
- d) The Fortran Preprocessor Definitions did not get converted correctly, need to add a “;” after each definition in the Project.
- a. Right click on the Fortran project
 - b. Select properties.
 - c. Select the Fortran->General page
 - d. Add semi-colons in “Preprocessor Definitions”
- e) In the VS6 Project I had added the CVF Fortran library “dformat.lib” to link to (Project->Settings; Link tab). Remove this from the INTEL/VS2008 project.
- a. Right click on the Fortran project
 - b. Select properties.
 - c. Select the Linker->Input
 - d. Remove the library “dformat.lib” from “Additional Dependencies”
- f) When Linking I get the errors:
- ```
1>nafxcwd.lib(afxmem.obj) : error LNK2005: "void * __cdecl operator new(unsigned int)" (??2@YAPAXI@Z) already defined in LIBCMTD.lib(new.obj)
1>nafxcwd.lib(afxmem.obj) : error LNK2005: "void __cdecl operator delete(void *)" (??3@YAXPAX@Z) already defined in LIBCMTD.lib(dbgdel.obj)
1>nafxcwd.lib(afxmem.obj) : error LNK2005: "void __cdecl operator delete[](void *)" (??_V@YAXPAX@Z) already defined in LIBCMTD.lib(delete2.obj)
```
- a. This is because of the order that the libraries are linked:  
<http://support.microsoft.com/kb/148652>
  - b. Right click on the Fortran project
  - c. Select properties.
  - d. Select the Linker->Input
  - e. For “Ignore Specific Library” add: nafxcwd.lib;Libcmtld.lib
  - f. For “Additional Dependencies add: nafxcwd.lib Libcmtld.lib ( now semi colon here)
- g) After making the above change I still get the error:
- ```
1>nafxcwd.lib(afxmem.obj) : error LNK2005: "void __cdecl operator delete(void *)" (??3@YAXPAX@Z) already defined in Libcmtld.lib(dbgdel.obj)
```
- a. Not sure why I get this error, so use /FORCE on the command line to force the code to link.
 - b. Right click on the Fortran project
 - c. Select properties.
 - d. Select the Linker->Command Line
 - e. Add /FORCE to the command line to force a link.
- h) For the error:
- ```
error LNK2019: unresolved external symbol _SIGNAL@12 referenced in function _TTYINT
```

add:

use IFPORT to the Fortran program

- i) For the error:  
error LNK2019: unresolved external symbol \_GETENV@16 referenced in  
function \_MCNP

add:

use IFPORT to the Fortran program

- j) For the error:  
error LNK2019: unresolved external symbol \_ETIME@4 referenced in  
function \_SECND

add:

use IFPORT to the Fortran program

- k) If the compiler then complains “Please specify the name of the executable file to be used for the debug session:” then Select browse and browse to the location of the executable (not sure why it can not find it on its own).

## Create by Hand:

If you decide you do not want to do the automatic update, you can do the conversion by hand. Below are the options I needed to set to get my code to work with the new compilers. Some of these options are specific to my program, but others are generic to the conversion process.

1. Create new project Fortran->Windowing Application->Empty Project
2. Set the name and directory (Name: and Location: )
3. Create directories and add files to the Fortran Project
4. Under Fortran->General
  - a. Additional Include Directories:  
**F:\MCNPX\_projects\MCNPX\_SRC\MCNPX26F\v26f\src\include**
  - b. Preprocessor Definitions:  
**INTEL; VISCOLORS; VECAD; VISED; CINDER; DEC; XS64;  
MESHTAL; RADIOG; CEM; HISTP; SPABI; DFACT; INCL; PLOT;  
MC PLOT; GKSSIM; CHEAP**
5. Under Fortran->Preprocessor
  - a. Preprocess Source File:  
**Yes**
6. Under Fortran->Diagnostics
  - a. Generate Interface Blocks:  
**No**
  - b. Check Routine Interfaces:  
**No**
7. Under Fortran->Data
  - a. Local Variable Storage  
**All Variables Save (/Qsave)**
8. Under Fortran->External Procedures
  - a. Calling Convention  
**CVF(/iface:cvf)**
9. Under Fortran->Run-time
  - a. Check Array and String Bounds  
**No**
10. Under Linker->Input
  - a. Additional Dependencies  
**nafxwd.lib Libcmt.lib OPENGL32.LIB GLU32.LIB Htmlhelp.lib  
(nafxw.lib Libcmt.lib ... on Release)**
  - b. Ignore Specific Library  
**nafxwd.lib;Libcmt.lib  
(nafxw.lib;Libcmt.lib on Release)**
11. Under Linker->System
  - a. Heap Reserve Size  
**83886080**
  - b. Stack Reserve Size

**67108864**

- c. Enable Large Addresses  
**Support Addresses Larger than 2 GB (/LARGEADDRESSAWARE)**
12. Under Linker->Command Line
  - a. add  
**/MACHINE:I386 /FORCE**
1. File-> new-> project Visual C++->MFC->MFC DLL
  - a. Set Name: to  
**vislib**
  - b. Set Solution: to  
**Add to Solution**
  - c. Select OK, then Finish
2. Delete any .c and .h files that have been created
3. Add the c files to the project (Add->Existing Item).
4. Under General
  - a. Configuration Type  
**Static Library (.lib)**
  - b. Use of MFC  
**Use MFC in a Static Library**
  - c. Character Set  
**Use Multi-Byte Character Set**
5. Under Debugging
  - a. Command  
**F:\MCNPX\_projects\VISEDX\_23E\_VS2008\visedX23E\Debug\visedx23e.exe**
6. C/C++->Preprocessor (C/C++ will not show up if you have not added files)
  - a. Preprocessor Definitions  
**Remove\_USRDLL**
7. C/C++->Code Generation
  - a. Runtime Library  
**Multi-threaded Debug (/Mtd)**  
**(Multi-threaded /MT on Release)**
8. C/C++->Precompiled Headers
  - a. Create/Use Precompiled Header  
**Not Using Precompiled Headers**
9. Right click on the Fortran Project
  - b. Select Project Dependencies  
**Click on the "vislib" check box**
10. Right click on the Fortran Project and add a vised files directory
  - c. Add  
**Vised2.rc**

## The Rest of the Story:

1. File-> new-> project Visual C++->MFC->MFC Application
2. Set the name and directory (Name: and Location: )
3. Select OK, then Finish
4. Delete any .c and .h files that have been created
5. Add the c files to the project (Add->Existing Item).
6. Right click on the project, select properties
7. Under General
  - a. Use of MFC  
**Use MFC in a Static Library**
  - b. Character Set  
**Use Multi-Byte Character Set**
8. C/C++->Code Generation
  - a. Runtime Library  
**Multi-threaded Debug (/Mtd)**  
**(Multi-threaded /MT on Release)**
9. C/C++->Precompiled Headers
  - a. Create/Use Precompiled Header  
**Not Using Precompiled Headers**
10. Under Linker->General
  - a. Additional Library Directories  
**C:\Program Files\Intel\Compiler\Fortran\10.1.021\IA32\Lib**
11. Under Linker->Input
  - b. Additional Dependencies  
**OPENGL32.LIB GLU32.LIB Htmlhelp.lib**
12. Under Linker->System
  - c. Heap Reserve Size  
**83886080**
  - d. Stack Reserve Size  
**67108864**
  - e. Enable Large Addresses  
**Support Addresses Larger than 2 GB (/LARGEADDRESSAWARE)**
13. Select OK to exit the property pages
  
14. Create new project Fortran->Library->Static Library
  - a. Set Name: to  
**flib**
  - b. Set Solution: to  
**Add to Solution**
  - c. Select OK
15. Create directories and add files to the Fortran Project
16. Under Fortran->General
  - a. Additional Include Directories:  
**F:\MCNPX\_projects\MCNPX\_SRC\MCNPX26F\v26f\src\include**



- b. Preprocessor Definitions:  
**INTEL; VISCOLORS; VECAD; VISED; CINDER; DEC; XS64;  
MESHTAL; RADIOG; CEM; HISTP; SPABI; DFACT; INCL; PLOT;  
MCPLOT; GKSSIM; CHEAP**
- 17. Under Fortran->Preprocessor
  - a. Preprocess Source File:  
**Yes**
- 18. Under Fortran->Diagnostics
  - a. Generate Interface Blocks:  
**No**
  - b. Check Routine Interfaces:  
**No**
- 19. Under Fortran->Data
  - a. Local Variable Storage  
**All Variables Save (/Qsave)**
- 20. Under Fortran->External Procedures
  - a. Calling Convention  
**CVF(/iface:cvf)**
- 21. Under Fortran->Run-time
  - a. Check Array and String Bounds  
**No**
- 21 Right click on the C++ Project
  - a. Select Project Dependencies  
**Click on the “flib” check box**

## When Compiling without the Visual Editor

1. You must change Linker->System from Windows to Console

If you get this error:

There is no executable, change the C project form a .lib to a .exe

