

Checklist For C++ calling Fortran (Notes on sample C_calls_Fortran solution) with Visual Studio 2017 and Visual C++ and Intel Fortran 2018:

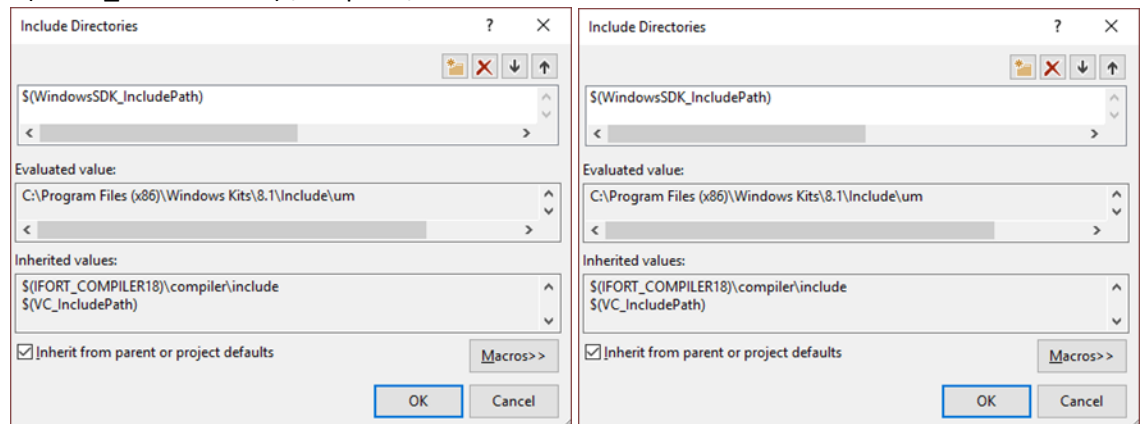
1) Global Visual Studio Settings for C interoperability with Fortran 2018:

a. Settings required for Intel Fortran 2018: (see

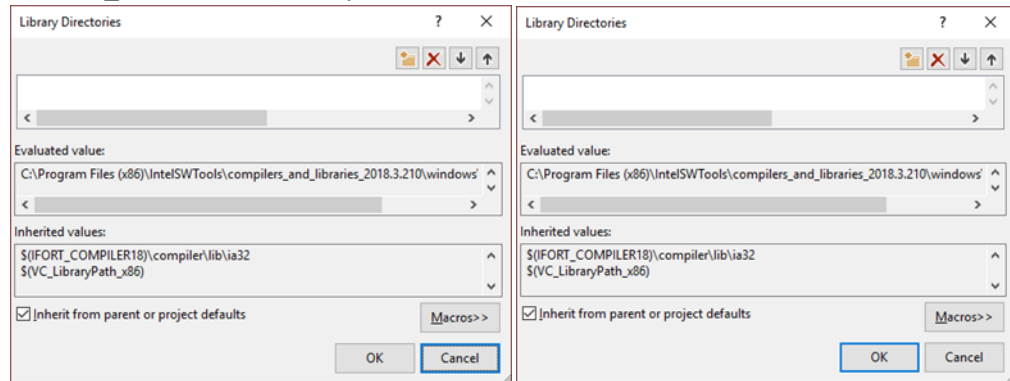
<https://software.intel.com/en-us/articles/configuring-visual-studio-for-mixed-language-applications>)

1. In Visual Studio, with a solution open that contains a C++ project, select View > Property Manager. If you do not see Property Manager under the View menu, you will find it under View > Other Windows. The Property Manager window will appear. Note that this is not Properties Window or Properties Pages.
2. Click on the triangles or + signs to expand the property tree under the Debug|Win32 configuration. See the image below.
3. Double click on Microsoft.Cpp.Win32.user
4. Select VC++ Directories
5. Click in the field to the right of "Include Directories"
6. Click the triangle that appears to the right and select <Edit...>
7. Click the New Line button or press Ctrl-Insert
8. In the new field that appears, type:
\$(IFORT_COMPILER[**vv**])\compiler\include
(remember to substitute for [vv] as described above)
9. Click OK
10. Click in the field to the right of "Library Directories"
11. Click the triangle that appears to the right and select <Edit...>
12. Click the New Line button or press Ctrl-Insert
13. In the new field that appears, type:
\$(IFORT_COMPILER[**vv**])\compiler\lib\ia32
(remember to substitute for [vv] as described above)
14. Click OK, OK
15. In the Visual Studio toolbar, select File > Save All

\$(IFORT_COMPILER18)\compiler\include

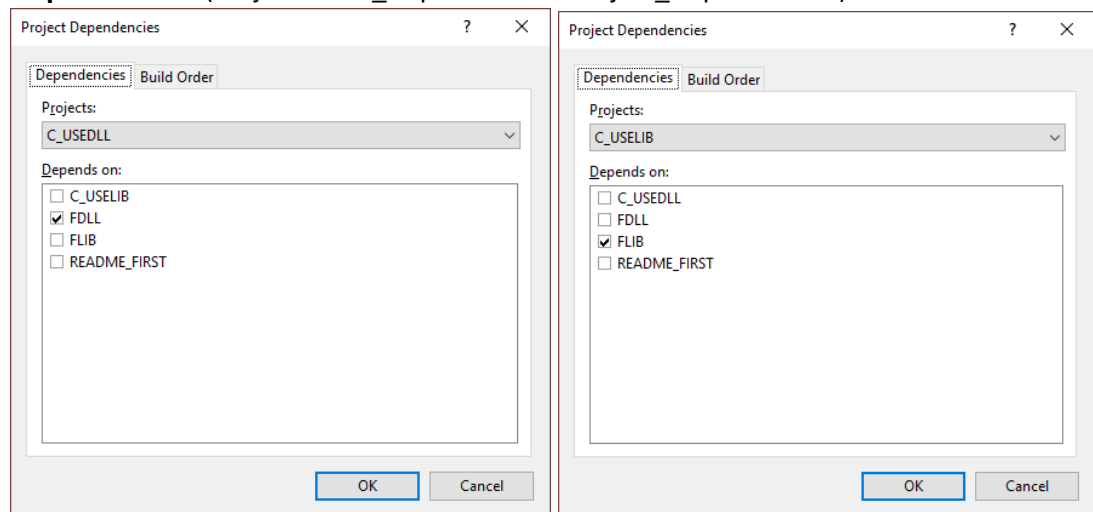


\$(IFORT_COMPILER18)\compiler\lib\ia32



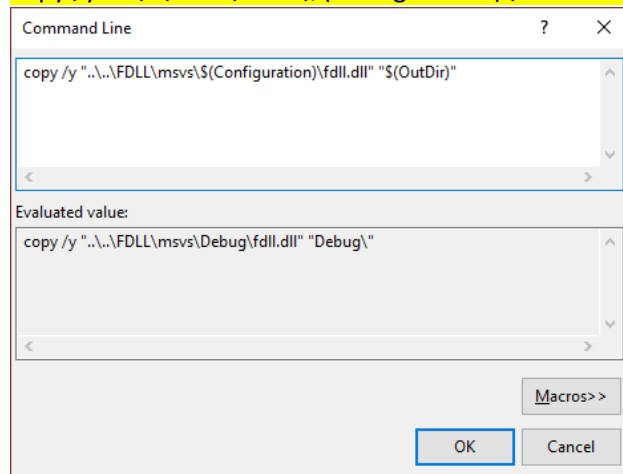
2) Project Settings: (Controls compilation and build order. Allows inheritance of project properties for a dependent project – See checkbox at the bottom of the dialog boxes above.)

a. Dependencies (Project>Build_Dependencies>Project_Dependencies)

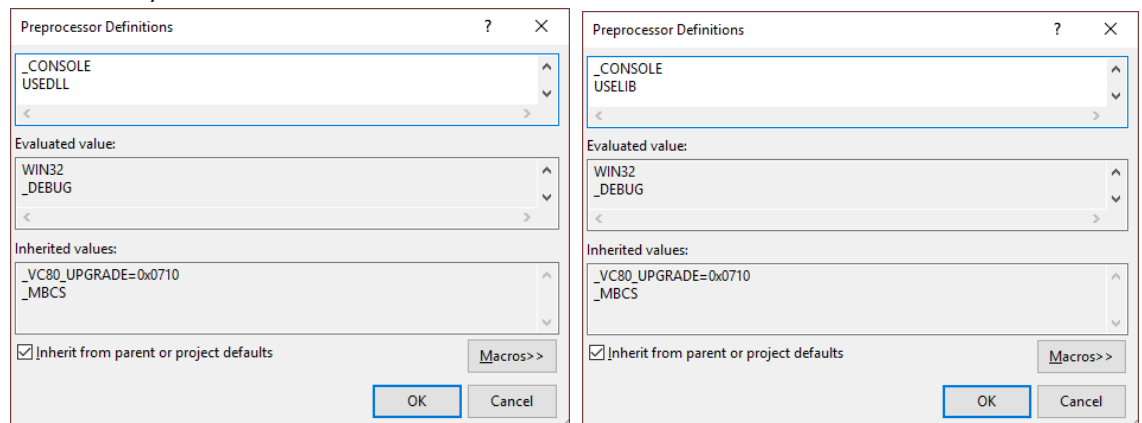


b. Copying the Fortran DLL to the executable file directory by Visual Studio (Project Properties Pages: Build_Events>PostBuild-Events)

copy /y "..\..\FDLL\msvs\\$(Configuration)\fdll.dll" "\$(OutDir)"



- c. **Additional symbols for debug control etc.** (C/C++>Preprocessor>...scroll..."USEDLL" or "USELIB")



- d. Xxx

3) Code elements:

- a. **C++: (CMAIN.CPP)**

- i. #include "ISO_Fortran_binding.h"
- ii. Fortran subroutine interface: include the Fortran C Interoperability declarations in front of the Fortran subroutine interface. The "C" attribute prevents C++ name mangling (predefined macro __cplusplus)

```
#ifdef __cplusplus  
extern "C"  
#endif
```

- iii. Fortran subroutine interface: If using a Fortran DLL, add the "dllimport" storage-class attribute or specification (see also dllexport) in the interface

```
#ifdef USEDLL  
__declspec(dllimport)  
#endif
```

- iv. Fortran subroutine interface (Passing parameters/descriptors to Fortran):

```
void fsub
```

```
(int INT_ARG,
 CFI_cdesc_t *STR_IN,
 CFI_cdesc_t *STR_OUT
 );
```

- v. Initialization or declaration of descriptors to pass strings between (Provided by the Intel Compiler or Studio - see "Extended interoperability with C by Reinhold Bader" – (0)=Scalar, (n)=n dimensional array)

```
CFI_CDESC_T(0) instring_desc, outstring_desc;
// Ignore warning for zero-sized array when using it with Visual C++
```

- vi. Constructor function to initialize members of the created Descriptor (the error codes are defined in "ISO_Fortran_binding.h"):

```
CFI_establish(
 (CFI_cdesc_t *) &instring_desc, // Descriptor
 &instring, // Base address
 CFI_attribute_other, // Not allocatable or pointer
 CFI_type_char, // Character type
 strlen(instring), // Element size
 0, // Number of dimensions (0 = scalar)
 NULL // Extents (not used here)
 );
```

- vii. Fortran subroutine (Passing descriptors to Fortran) call

```
/* Call Fortran routine, passing descriptors. "intarg" is passed by value */
fsub(intarg,(CFI_cdesc_t *)&instring_desc,(CFI_cdesc_t *)&outstring_desc);
```

b. Fortran: (Subroutine FSUB.F90)

- i. BIND(C) for "interoperability with C": (Since the optional NAME= specifier was not used, the external name is downcased to "fsub". Any platform-specific name decoration is automatically added to match the companion C processor.)

```
SUBROUTINE FSUB (INT_ARG, STR_IN, STR_OUT) BIND(C)
USE, INTRINSIC :: ISO_C_BINDING
```

- ii. Export DLL_ROUT if fortran is packaged as a DLL (_DLL symbol is automatically created if the project was created as DLL. See FDLL project properties General>Configuration_Type)

```
!DEC$ IF DEFINED (_DLL)
!DEC$ ATTRIBUTES DLLEXPORT :: FSUB
!DEC$ END IF
```

- iii. xx

c. xxx

4) Test: C_calls_Fortran (C:\AArea\ik18a\MixedPrgmTest1\C-To-Fort\C_calls_Fortran)

- a. **Compiler/Linker Output:** Build>Clean_Solution, Build>Rebuild_Solution

```
1>----- Skipped Rebuild All: Project: README_FIRST, Configuration: Debug Win32 -----
1>Project not selected to build for this solution configuration
2>----- Rebuild All started: Project: FDLL, Configuration: Debug Win32 -----
3>----- Rebuild All started: Project: FLIB, Configuration: Debug Win32 -----
```

```
2>Deleting intermediate files and output files for project 'FDLL', configuration
'Debug|Win32'.
3>Deleting intermediate files and output files for project 'FLIB', configuration
'Debug|Win32'.
3>Compiling with Intel(R) Visual Fortran Compiler 18.0.3.210 [IA-32]...
2>Compiling with Intel(R) Visual Fortran Compiler 18.0.3.210 [IA-32]...
3>FSUB.F90
2>FSUB.F90
2>Linking...
3>Creating library...
3>
3>Build log written to
"file:///C:/AArea/ik18a/MixedPrgmTest2/C_calls_Fortran/FLIB/msvs/Debug/BuildLog.ht
m"
3>FLIB - 0 error(s), 0 warning(s)
4>----- Rebuild All started: Project: C_USELIB, Configuration: Debug Win32 -----
2>Creating library Debug/FDLL.lib and object Debug/FDLL.exp
2>Embedding manifest...
2>
2>Build log written to
"file:///C:/AArea/ik18a/MixedPrgmTest2/C_calls_Fortran/FDLL/msvs/Debug/BuildLog.ht
m"
2>FDLL - 0 error(s), 0 warning(s)
5>----- Rebuild All started: Project: C_USEDLL, Configuration: Debug Win32 -----
4>CMAIN.CPP
5>CMAIN.CPP
5>C:\Program Files
(x86)\IntelSWTools\compilers_and_libraries_2018.3.210\windows\compiler\include\IS
O_Fortran_binding.h(156): warning C4200: nonstandard extension used : zero-sized
array in struct/union
5>    Cannot generate copy-ctor or copy-assignment operator when UDT contains a
zero-sized array
5>..\..\src\CMAIN.CPP(87): warning C4200: nonstandard extension used : zero-sized
array in struct/union
5>    Cannot generate copy-ctor or copy-assignment operator when UDT contains a
zero-sized array
4>C:\Program Files
(x86)\IntelSWTools\compilers_and_libraries_2018.3.210\windows\compiler\include\IS
O_Fortran_binding.h(156): warning C4200: nonstandard extension used : zero-sized
array in struct/union
4>    Cannot generate copy-ctor or copy-assignment operator when UDT contains a
zero-sized array
4>..\..\src\CMAIN.CPP(87): warning C4200: nonstandard extension used : zero-sized
array in struct/union
```

```

4> Cannot generate copy-ctor or copy-assignment operator when UDT contains a
zero-sized array
4> Creating library Debug\C_USELIB.lib and object Debug\C_USELIB.exp
5>USEDLL.vcxproj ->
C:\AArea\ik18a\MixedPrgmTest2\C_calls_Fortran\USEDLL\msvs\Debug\C_USEDLL.exe
4>USELIB.vcxproj ->
C:\AArea\ik18a\MixedPrgmTest2\C_calls_Fortran\USELIB\msvs\Debug\C_USELIB.exe
4>Done building project "USELIB.vcxproj".
5> 1 file(s) copied.
5>Done building project "USEDLL.vcxproj".
===== Rebuild All: 4 succeeded, 0 failed, 1 skipped =====

```

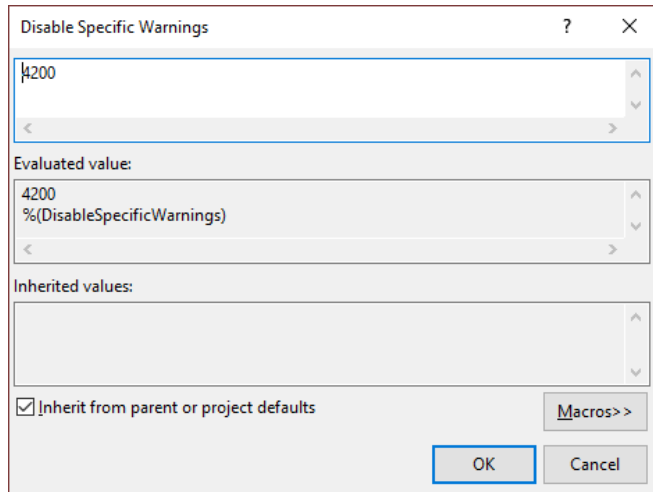
b. Errors or warnings (See section 5. Miscellaneous for disabling warning codes)

| Code | Description | Project | File | Line |
|-------|---|----------|-----------------------|------|
| C4200 | nonstandard extension used : zero-sized array in struct/union | C_USELIB | ISO_Fortran_binding.h | 156 |
| C4200 | nonstandard extension used : zero-sized array in struct/union | C_USEDLL | ISO_Fortran_binding.h | 156 |
| C4200 | nonstandard extension used : zero-sized array in struct/union | C_USEDLL | CMAIN.CPP | 87 |
| C4200 | nonstandard extension used : zero-sized array in struct/union | C_USELIB | CMAIN.CPP | 87 |

c. xxx

5) Miscellaneous:

a. **"Disable Specific Warnings"** (Enter only "4200" NOT "C4200". Correct entry does not prevent the warnings.)



b. xx