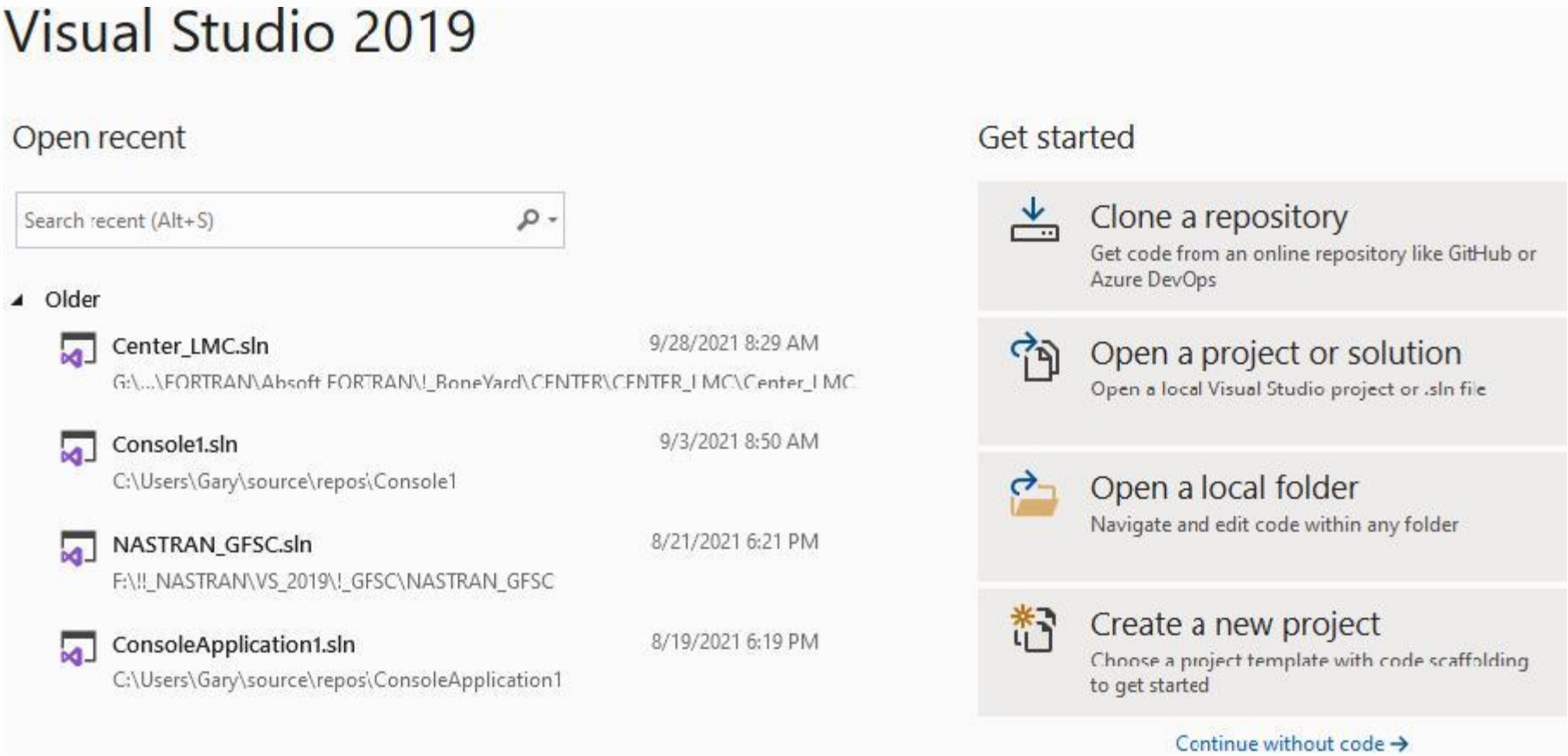
2021 Aug 30 OneAPI Toolkits and Visual Studio 2019 Tips Updated to VS 2022 no changes Updated to 2023 April Option on Compiler

Gary L. Fox LM Skunk Works

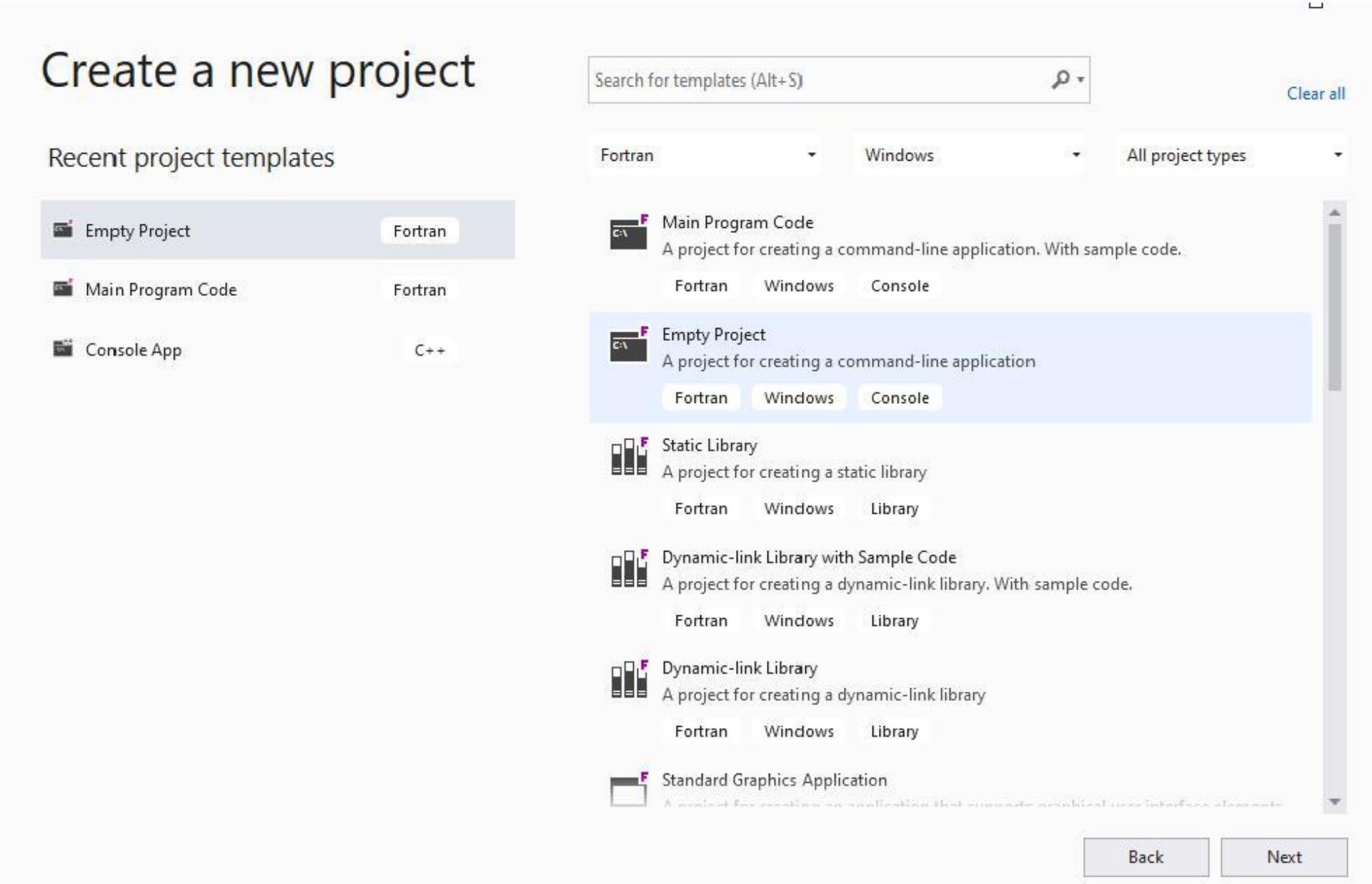
MS Visual Studio

This interface is a lot like the old Compaq but enough different to make it very frustrating to learn

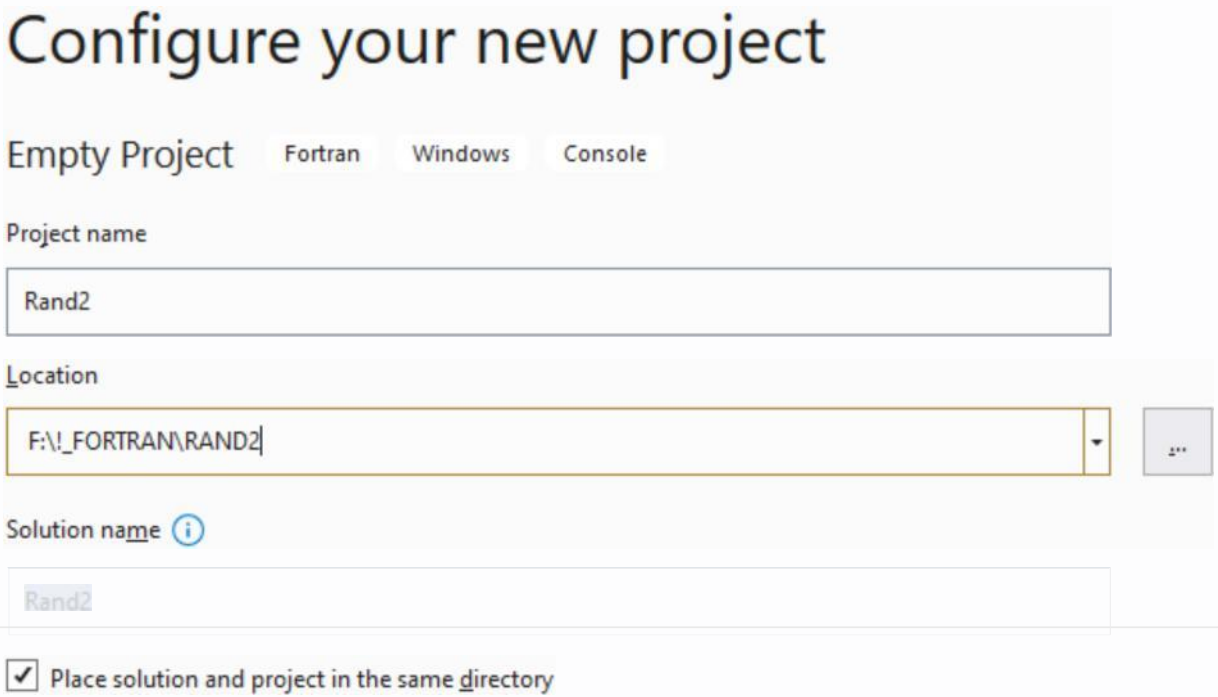
This is the open screen
click Create a new Project



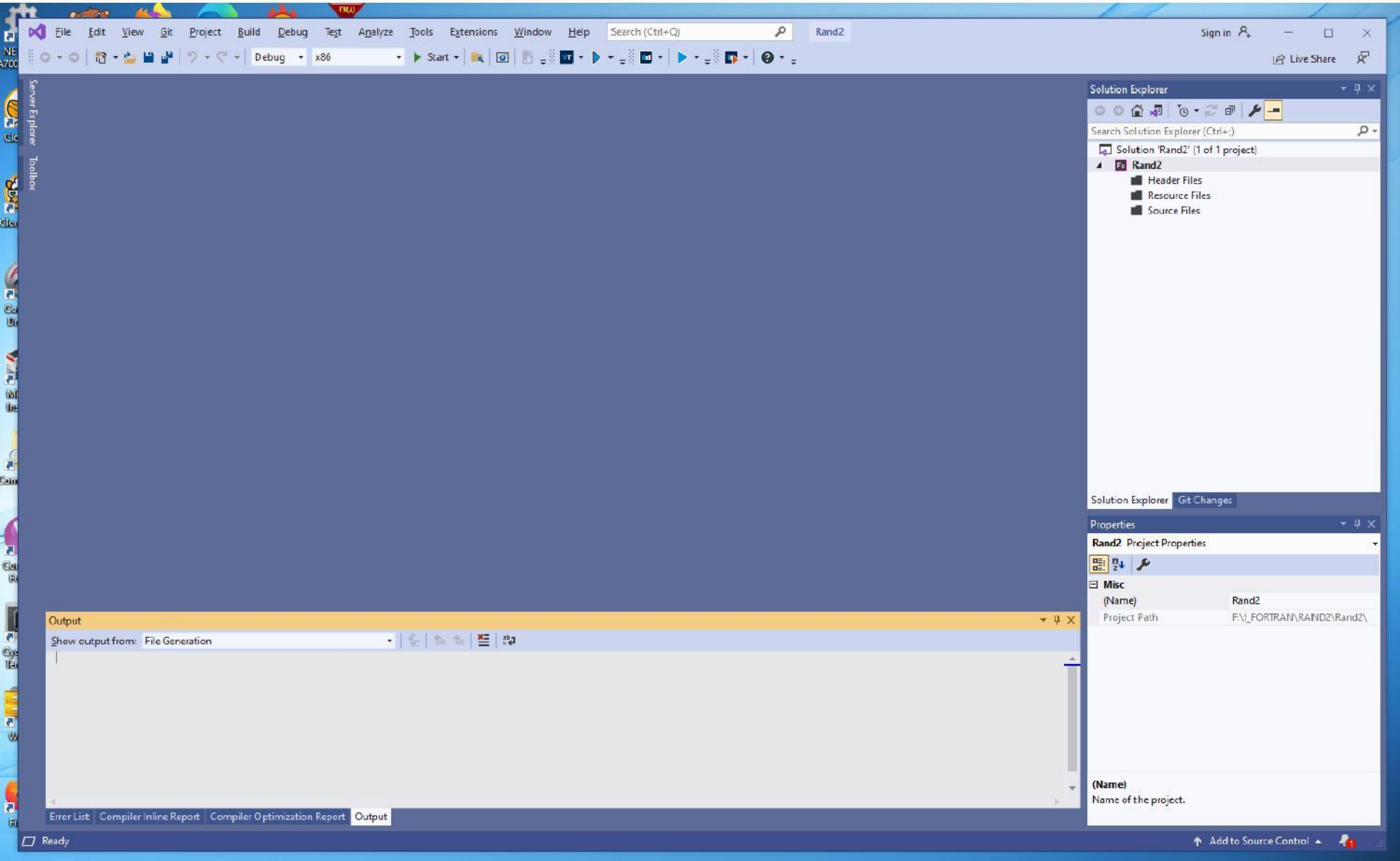
Select Empty Project
Fortran Windows Console



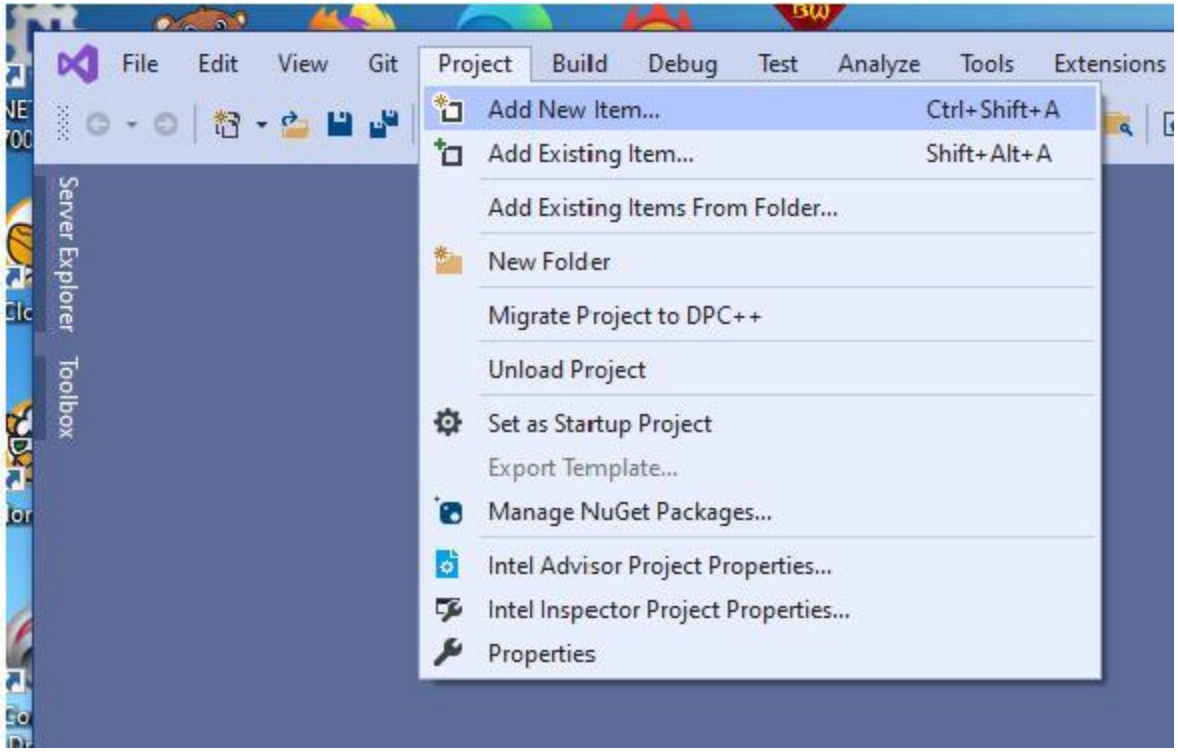
A solution in VS may have multiple projects.
To date I have only one so the box is checked to put the solution and project in the same folder.
Note: Rand2 is a one dof random response program used for this demo.



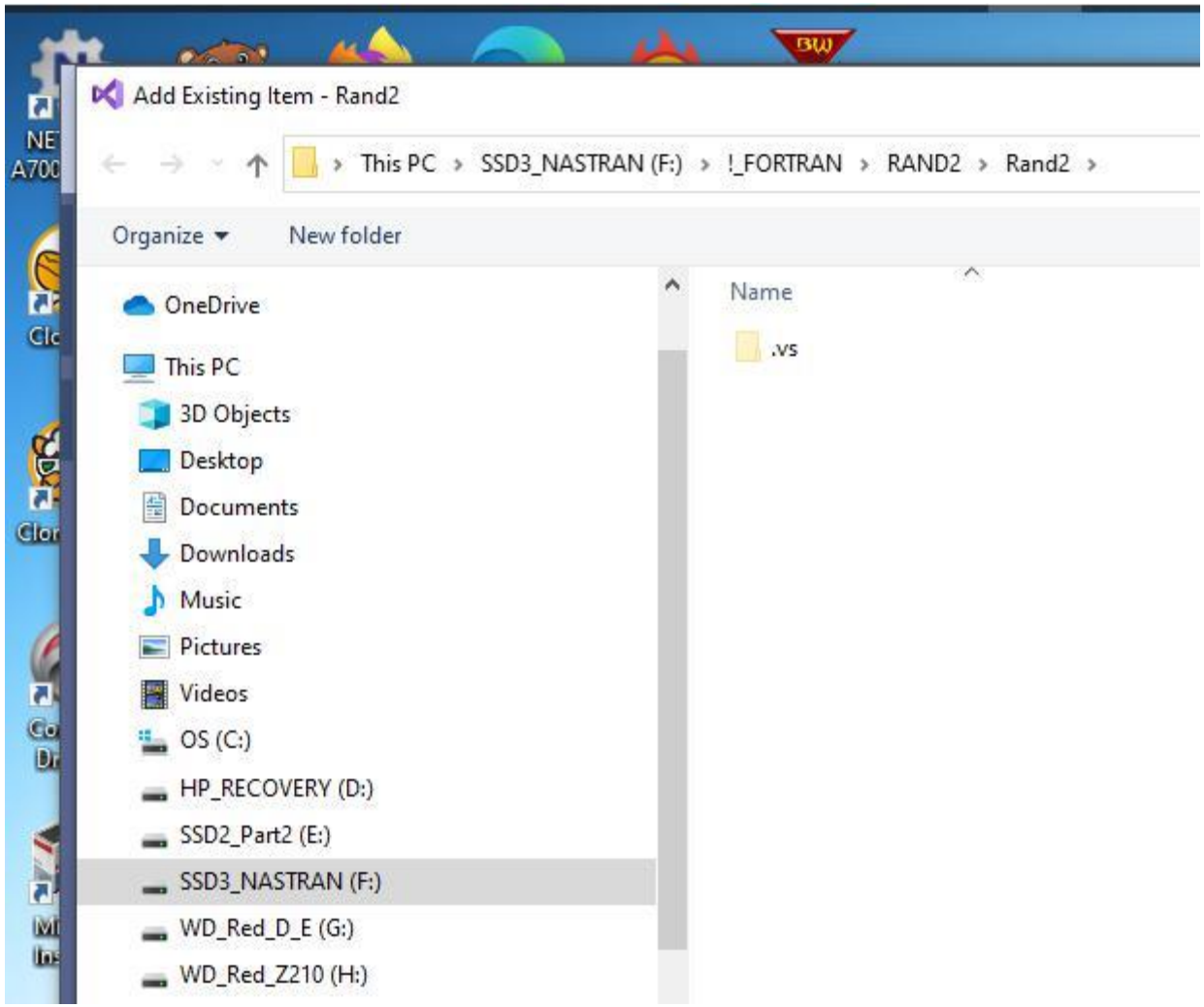
This is the open screen. Now need to add the Rand2 fortran file to the project.



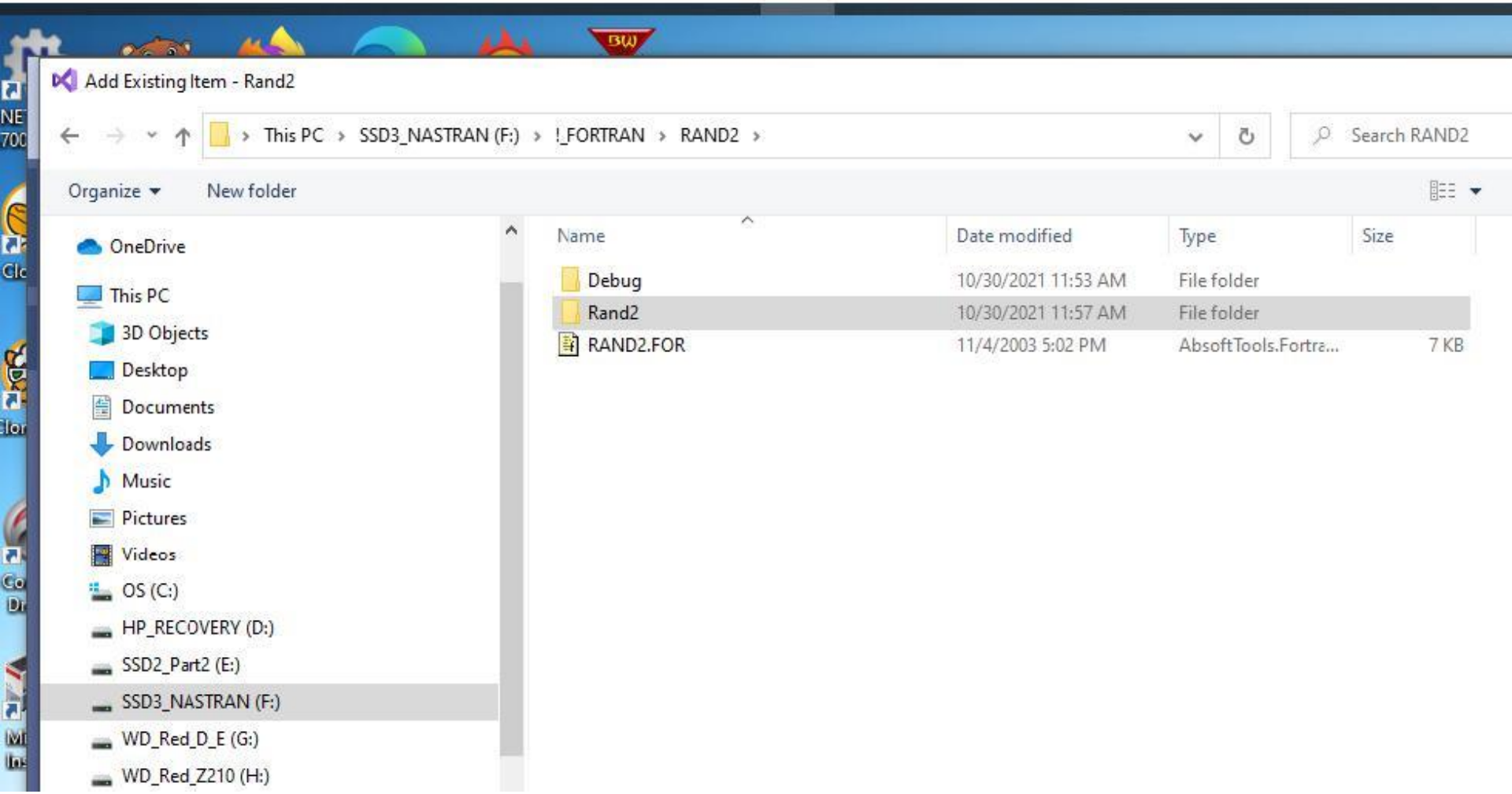
Add existing Item



Project added Rand2 folder below the RAND2 pre-existing folder



Select the RAND2 folder where the code is

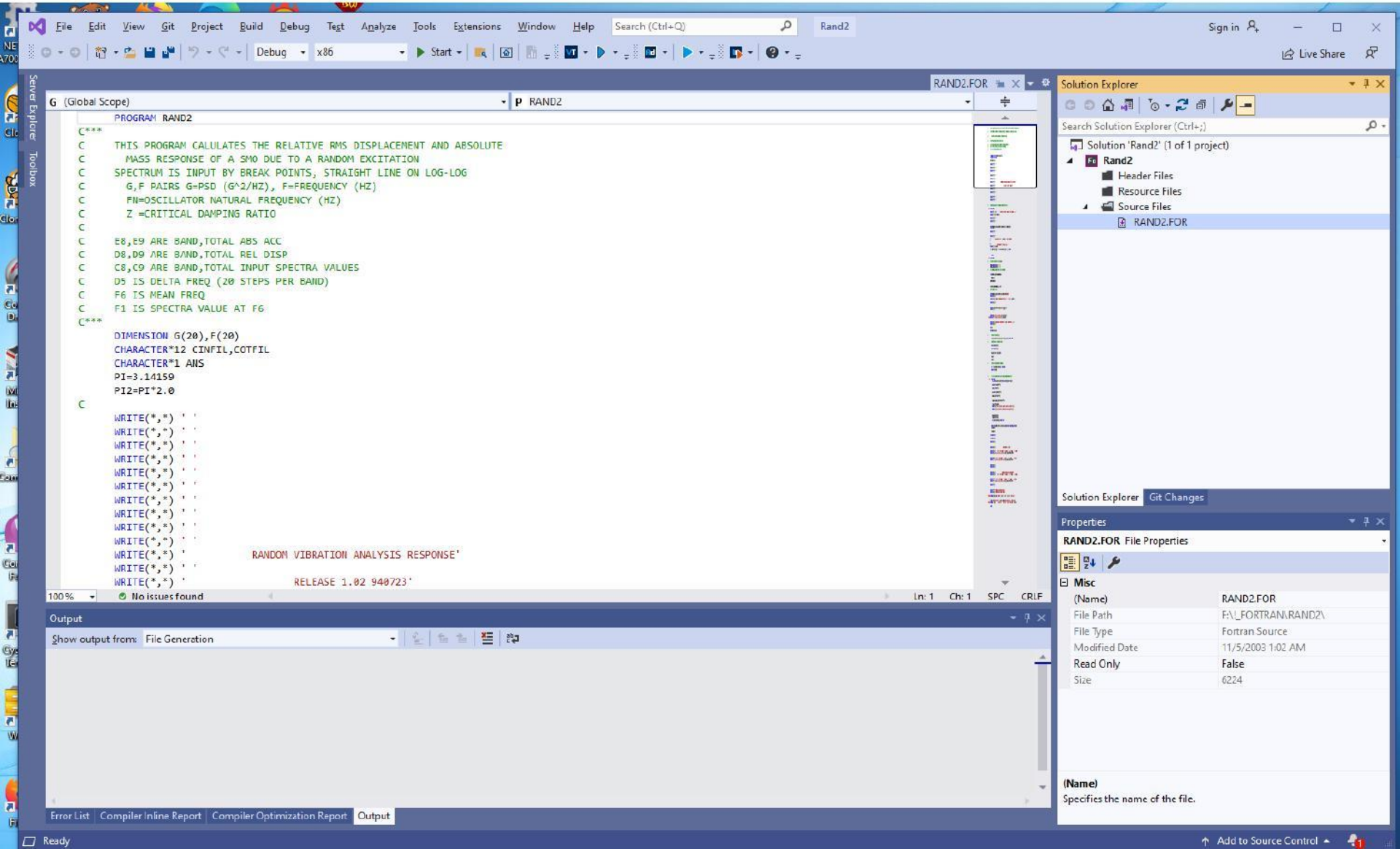


Notice that RAND2.FOR is now listed in the "Solution Explorer" box – Case Sensitive!!!!

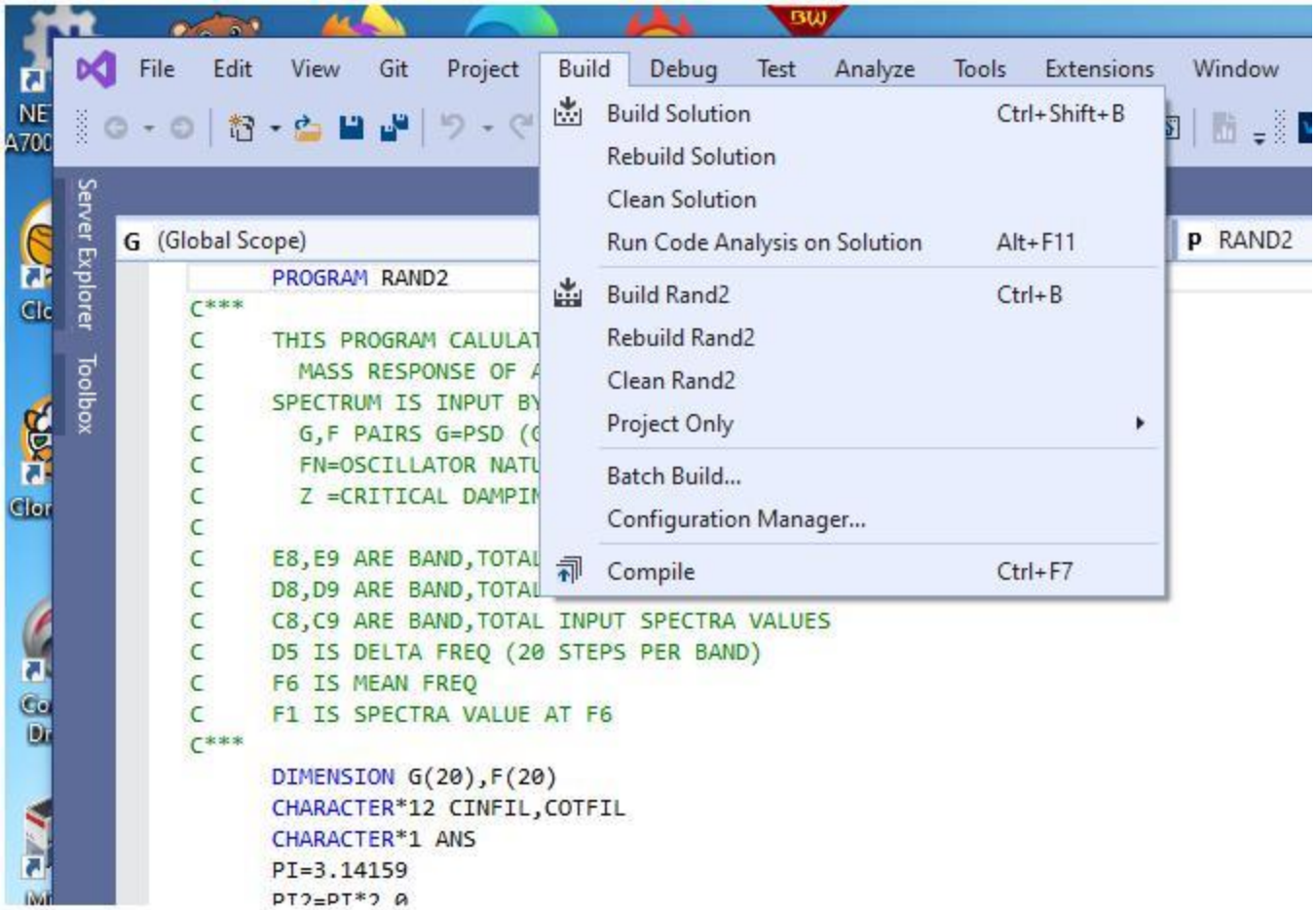
Note also the Project Path – that is where the compiled code will default for data.

I believe that if the "Use IFORT" directive is used in the main program, then the SetEvnironment command may work.

Click on any file in the list, in this case RAND2, and the code appears.



I usually compile at this point



Compiled without errors

```
      PROGRAM RAND2
C***
C     THIS PROGRAM CALULATES THE RELATIVE RMS DISPLACEMENT AND ABSOLUTE
C       MASS RESPONSE OF A SMO DUE TO A RANDOM EXCITATION
C     SPECTRUM IS INPUT BY BREAK POINTS, STRAIGHT LINE ON LOG-LOG
C       G,F PAIRS G=PSD (G^2/HZ), F=FREQUENCY (HZ)
C       FN-OSCILLATOR NATURAL FREQUENCY (HZ)
C       Z =CRITICAL DAMPING RATIO
C
C     E8,E9 ARE BAND,TOTAL ABS ACC
C     D8,D9 ARE BAND,TOTAL REL DISP
C     C8,C9 ARE BAND,TOTAL INPUT SPECTRA VALUES
C     D5 IS DELTA FREQ (20 STEPS PER BAND)
C     F6 IS MEAN FREQ
C     F1 IS SPECTRA VALUE AT F6
C***
      DIMENSION G(20),F(20)
      CHARACTER*12 CINFIL,COTFIL
      CHARACTER*1 ANS
      PI=3.14159
      PI2=PI*2.0
C
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) '          RANDOM VIBRATION ANALYSIS RESPONSE'
      WRITE(*,*) ' '
      WRITE(*,*) '                 RELEASE 1.02 940723'
```

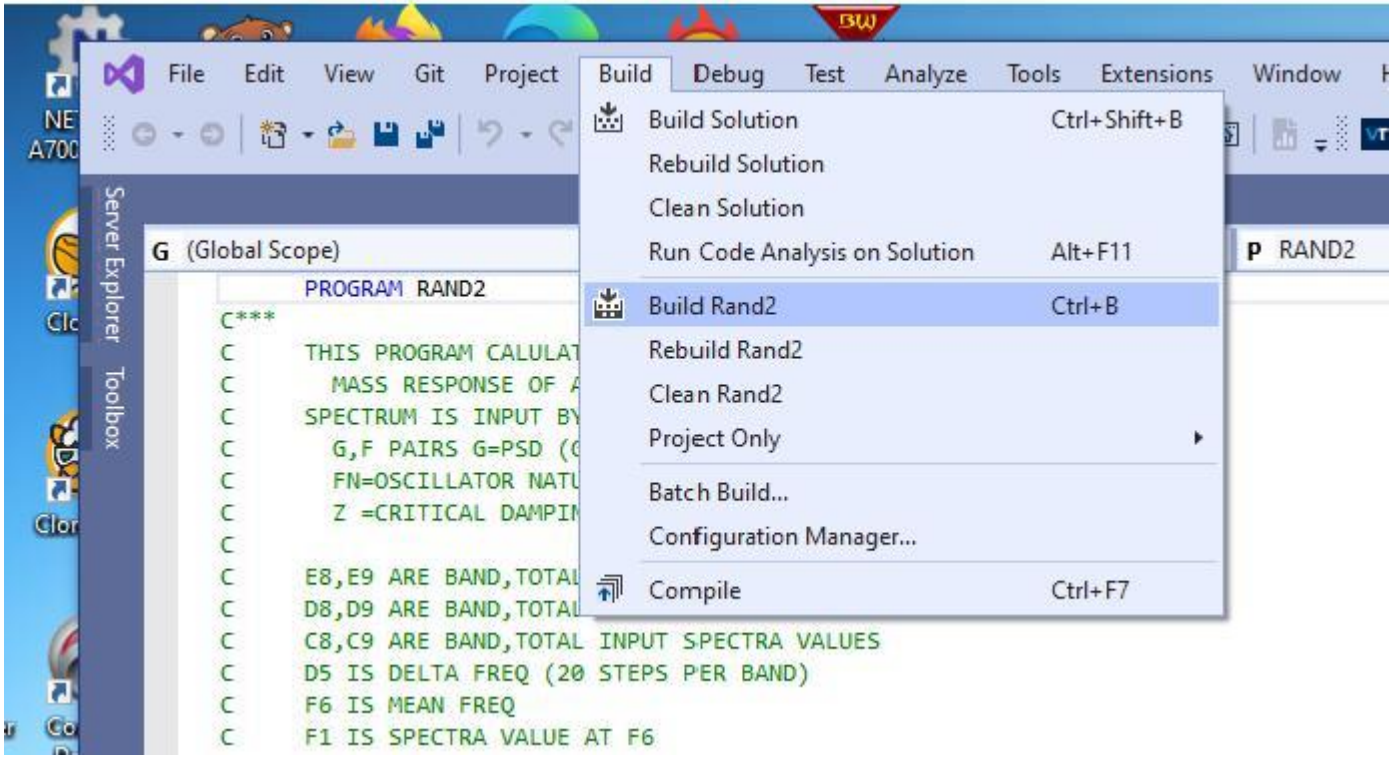
```
------ Build started: Project: Rand2, Configuration: Debug|Win32 ------

Compiling with Intel® Fortran Compiler Classic 2021.3.0 [IA-32]...
RAND2.FOR

Build log written to  "file:///F:\!_FORTRAN\RAND2\Rand2\Debug\BuildLog.htm"
Rand2 - 0 error(s), 0 warning(s)


---------------------- Done ----------------------
```

Build solution – this invokes the linker, so unresolved externals are found here



1 succeeded is good news, we have an executable now. By default is uses dlls, so need to make a “Static” program.

```
WRITE(*,*) ' '
WRITE(*,*) ' '
WRITE(*,*) ' '
WRITE(*,*) ' '
WRITE(*,*) '          RANDOM VIBRATION ANALYSIS RESPONSE'
WRITE(*,*) ' '
WRITE(*,*) '                RELEASE 1.02 940723'
```

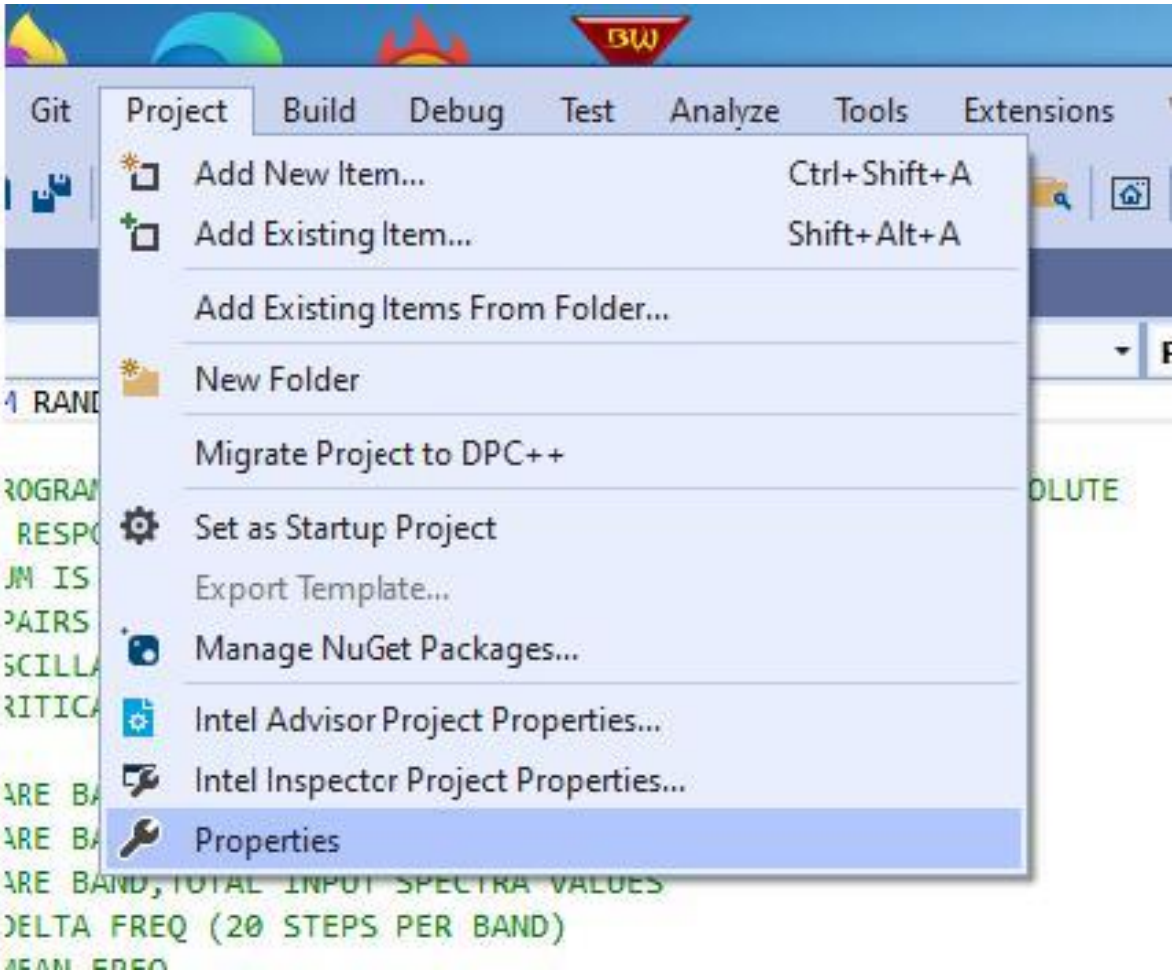
100 % No issues found

Output

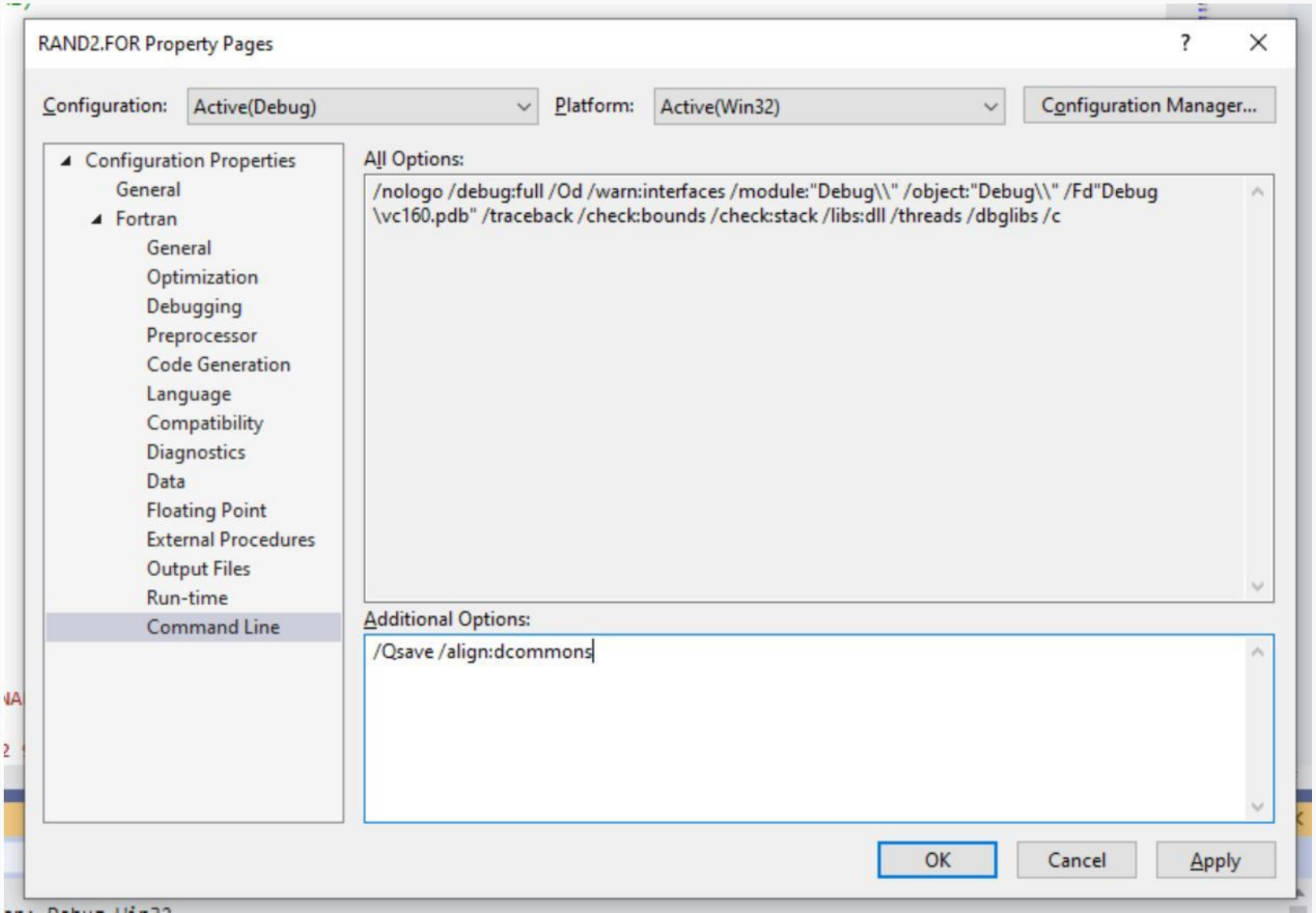
Show output from: Build

```
Build started...
1>------ Build started: Project: Rand2, Configuration: Debug Win32 ------
1>Linking...
1>Embedding manifest...
1>
1>Build log written to  "file://F:\! FORTRAN\RAND2\Rand2\Debug\BuildLog.htm"
1>Rand2 - 0 error(s), 0 warning(s)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Select Project Properties
Note: No Project Name, must have one –see later slides



/Qsave saves variables in a subroutine, increases memory
/align:dcommons makes sure 64 bit variables are correctly aligned
(NASTRAN had to have these to compile correctly, as well as
/check:none turn off bounds check since some older FORTRAN used dimension of (1) and later used more – a part of older FORTAN that makes common an issue with the "Elites".



Note: *Debug Multithreaded…* is an option on the Runtime Library tab

JSerra
Novice

07-03-2021 02:43 PM
276 Views

✓ Problem solved: in Project > Properties > Cinfig Properties > Fortran > Libraries > Runtime Library, I changed from *Multithread DLL (/libs:dll /threads)* to *Debug Multithreaded (/libs:static /threads /dbglibs)*. Then the .exe file runs normally and generates the intended file.

I hope for my (basic) use of VS this does not have collateral effects.

EDIT: see @Steve_Lionel 's warning regarding side effects.

View solution in original post

Translate

0 Kudos

Copy link | Share | Reply

Command Prompt (2) - rand2

```
Microsoft Windows [Version 10.0.19043.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gary>f:

F:\>cd F:\!_FORTRAN\RAND2

F:\!_FORTRAN\RAND2>dir
 Volume in drive F is SSD3_NASTRAN
 Volume Serial Number is 1C24-1G17

 Directory of F:\!_FORTRAN\RAND2

10/30/2021  01:48 PM    <DIR>          .
10/30/2021  01:48 PM    <DIR>          ..
10/30/2021  11:53 AM    <DIR>          Debug
11/04/2003  06:02 PM               110 FLAT_1.GIN
11/04/2003  06:03 PM            18,639 FLAT_1.ROT
11/04/2003  05:37 PM                95 FLAT_2.GIN
11/04/2003  06:09 PM               120 HDBK_1.GIN
11/04/2003  06:09 PM            20,446 HDBK_1.ROT
10/30/2021  12:11 PM    <DIR>          Rand2
09/03/2016  08:35 AM           544,837 RAND2 - Copy.
11/04/2003  05:52 PM             3,747 RAND2.DSP
11/04/2003  05:52 PM               533 RAND2.DSW
10/30/2021  01:48 PM            67,072 RAND2.exe
11/04/2003  06:02 PM             6,224 RAND2.FOR
11/04/2003  06:57 PM            41,984 RAND2.ncb
11/04/2003  06:57 PM            48,640 RAND2.OPT
11/04/2003  06:02 PM             1,118 RAND2.PLG
11/04/2003  07:16 PM               126 STAG123A.GIN
11/04/2003  07:16 PM            20,446 STAG123A.ROT
11/04/2003  07:08 PM               110 STAGE23A.GIN
11/04/2003  07:09 PM            16,832 STAGE23A.ROT
11/04/2003  07:00 PM               101 STAGE_1.GIN
11/04/2003  07:02 PM               101 STAGE_1A.GIN
11/04/2003  07:02 PM            15,025 STAGE_1A.ROT
07/23/1994  06:00 PM                72 TEST1.GIN
11/04/2003  05:14 PM            13,204 TEST1.ROT
              22 File(s)        819,582 bytes
               4 Dir(s)  3,858,188,468,224 bytes free

F:\!_FORTRAN\RAND2>rand2
```

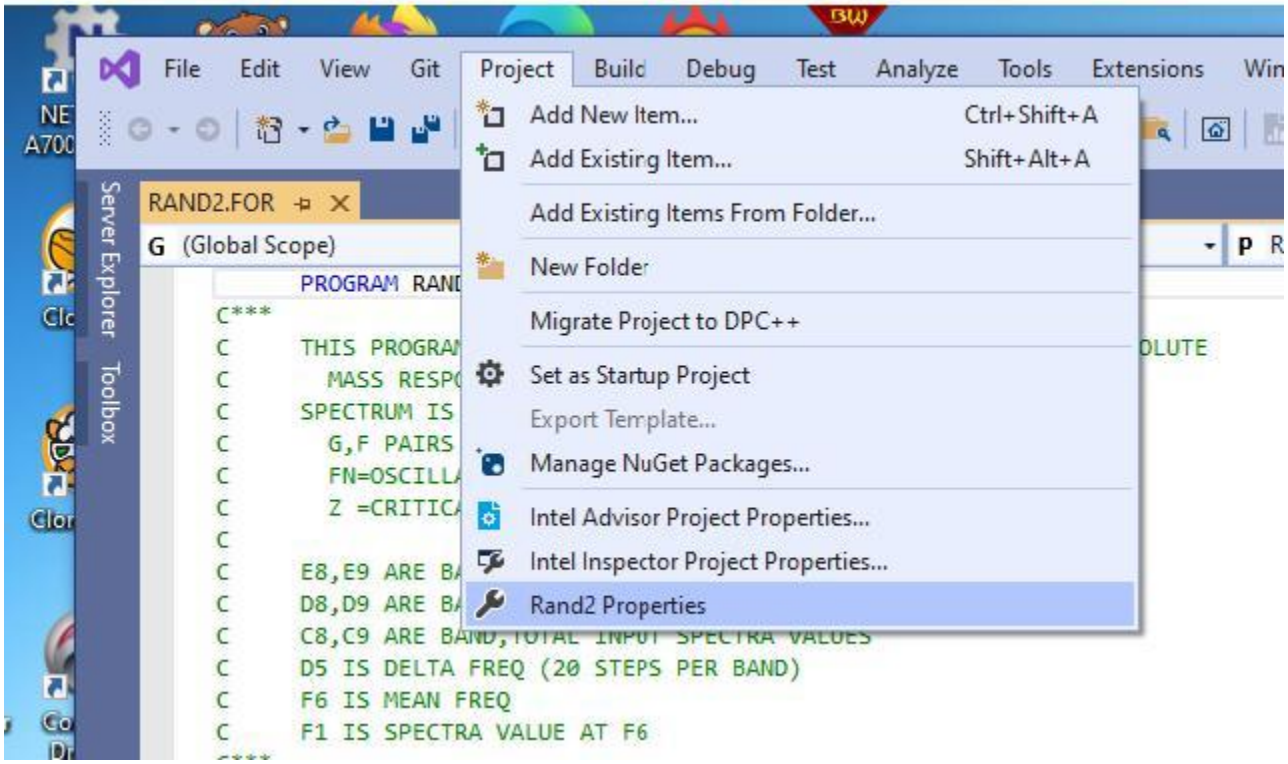
This is what happens without using a “static” library – see next slide

RAND2.exe - System Error

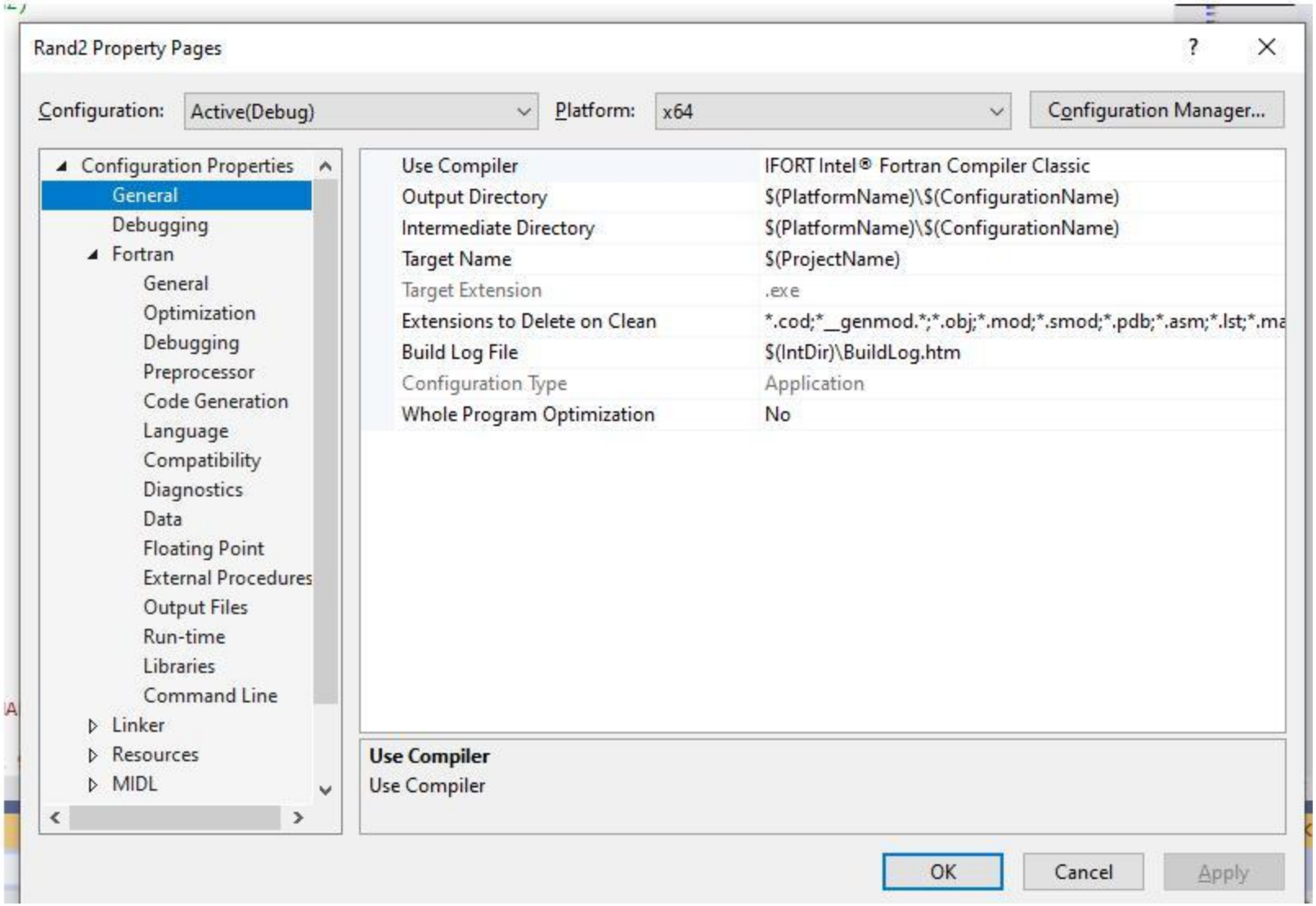
The code execution cannot proceed because libifcoremdd.dll was not found. Reinstalling the program may fix this problem.

OK

Make sure that you see the Project name (Rand2) before Properties
Sometimes this does not show up – wtf?



Need to see Libraries under Fortran



Center_LMC Property Pages ? X

Configuration Manager...

Configuration: Active(Debu

x64 V

- Configuration
  - Properties
  - General
  - Debugging
- Fortran
  - General
  - Optimization
  - Debugging
  - Preprocessor
  - Code
  - Generation
  - Language
  - Compatibility
  - Diagnostics
  - Data
  - Floating Point
  - External
  - Procedures Output
  - Libraries
  - Command Line
- Linker
- Resources
- MIDL

Runtime Library

Use Common Windows Libraries

Use Portlib Library

Use Intel Math Kernel Library

Disable Default Library Search Rules

Disable OBJCOMMENT Library Names in

Debug Multithreaded (ilibsstatic /threads /dbglibs)

Multithreaded

Multithread DLL Vlibs:dll /threads)

QuickWin (/libs:gwin)

Standard Graphics (ilibs:qwins)

Debug Multithreaded (/libs:static /threads /dbglibs)

Debug Multithread DLL (/libs:dll /threads /dbglibs)

Debug QuickWin Vlibs:gwin /dbglibs)

Debug Standard Graphics Vlibs:gwins /dbglibs)

<inherit from project defaults>

Runtime Library

Specifies the runtime library for linking. (/libsgstaticidlligwinigwins}, /threads, /dbglibs)

OK Cancel Apply

Select libs static

Rand2 Property Pages

Configuration: Active(Debug) | Platform: x64 | Configuration Manager...

- Configuration Properties
  - General
  - Debugging
  - Fortran
    - General
    - Optimization
    - Debugging
    - Preprocessor
    - Code Generation
    - Language
    - Compatibility
    - Diagnostics
    - Data
    - Floating Point
    - External Procedures
    - Output Files
    - Run-time
    - Libraries
    - Command Line
  - Linker
  - Resources
  - MIDL

Runtime Library	Debug Multithreaded (/libs:static /threads /dbglibs)
Use Common Windows Libraries	
Use Portlib Library	
Use Intel Math Kernel Library	
Disable Default Library Search Rules	
Disable OBJCOMMENT Library Names in	

- Multithreaded
- Multithread DLL (/libs:dll /threads)
- QuickWin (/libs:qwin)
- Standard Graphics (/libs:qwins)
- Debug Multithreaded (/libs:static /threads /dbglibs)
- Debug Multithread DLL (/libs:dll /threads /dbglibs)
- Debug QuickWin (/libs:qwin /dbglibs)
- Debug Standard Graphics (/libs:qwins /dbglibs)
- <inherit from project defaults>

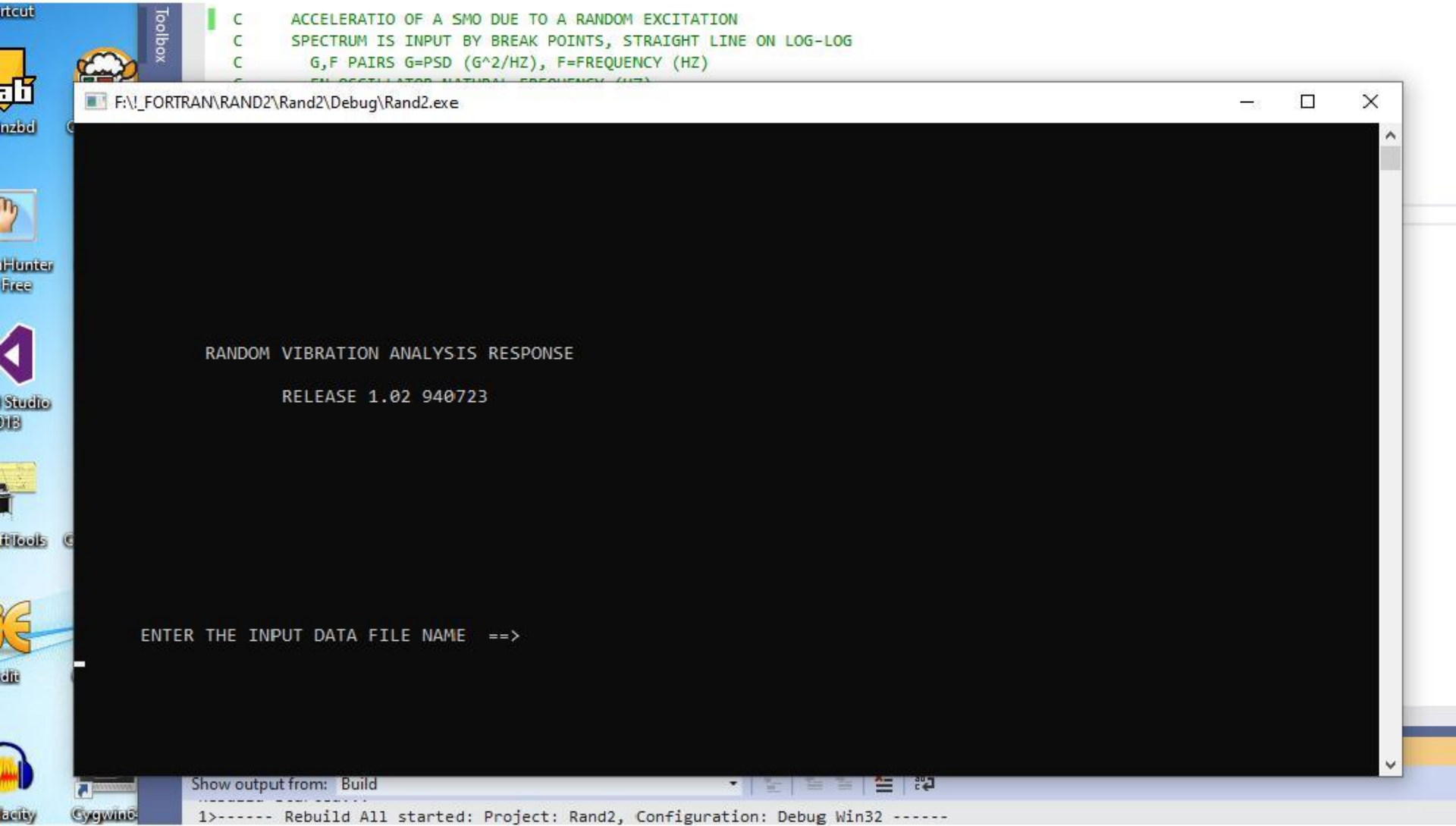
Runtime Library

Specifies the runtime library for linking. (/libs:{static|dll|qwin|qwins}, /threads, /dbglibs)

OK | Cancel | Apply

```
roject 'Rand2', configuration 'Debug|Win32'.
1.3.0 [IA-32]...
```

Finally got it to run – the systems wants to use dlls – had to repeat a few times for it to “take”



Release

Setting Data Type and Alignment

Alignment of data affects these kinds of variables:

- Those that are dynamically allocated. (Dynamically allocated data allocated with `ALLOCATE` is 8-byte aligned.)
- Those that are members of a data structure
- Those that are global or local variables
- Those that are parameters passed on the stack

For best performance, align data as follows:

- Align 8-bit data at any address.
- Align 16-bit data to be contained within an aligned four byte word.
- Align 32-bit data so that its base address is a multiple of four.
- Align 64-bit data so that its base address is a multiple of eight.
- Align 128-bit data so that its base address is a multiple of sixteen (8-byte boundaries).

Causes of Unaligned Data and Ensuring Natural Alignment

For optimal performance, make sure your data is aligned naturally. A natural boundary is a memory address that is a multiple of the data item's size. For example, a `REAL (KIND=8)` data item aligned on natural boundaries has an address that is a multiple of 8. An array is aligned on natural boundaries if all of its elements are so aligned.

All data items whose starting address is on a natural boundary are naturally aligned. Data not aligned on a natural boundary is called unaligned data.

Although the Intel® compiler naturally aligns individual data items when it can, certain Fortran statements can cause data items to become unaligned.

You can use the align command-option to ensure naturally aligned data, but you should check and consider reordering data declarations of data items within common blocks, derived-type structures, and record structures as follows:

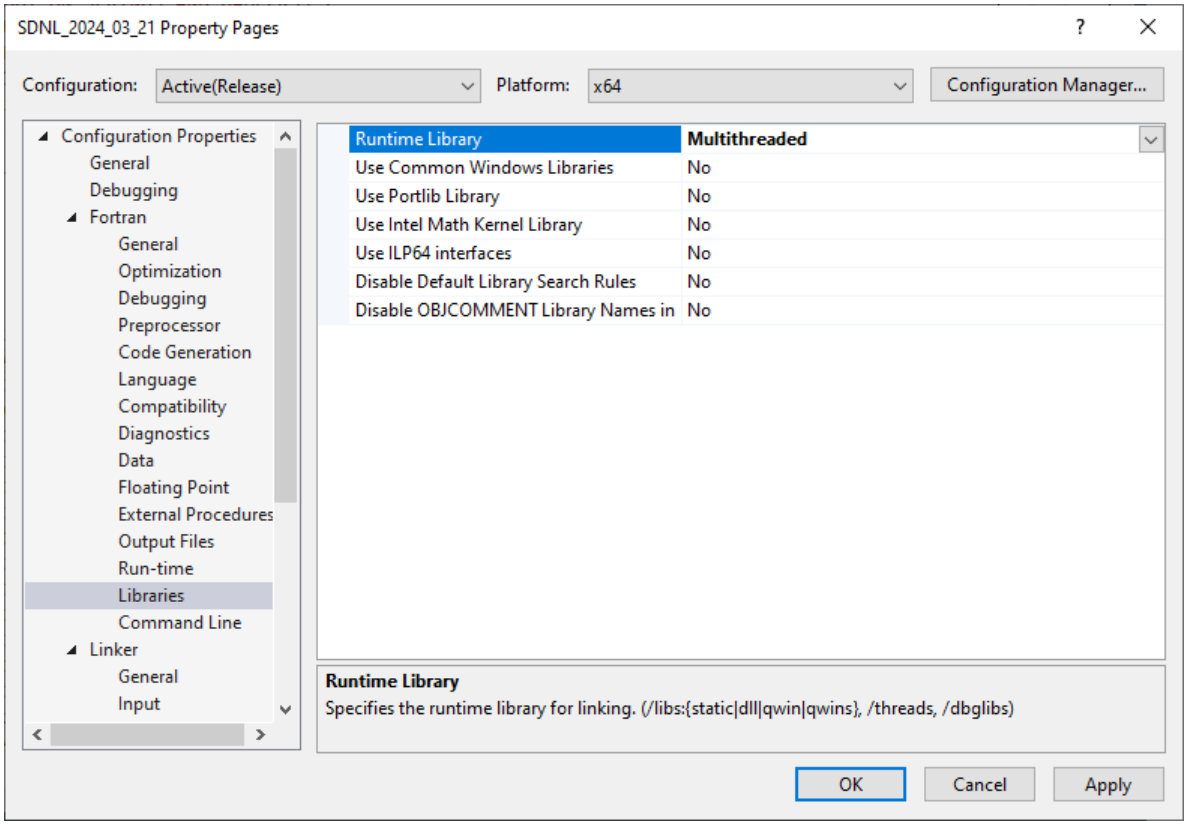
- Carefully specify the order and sizes of data declarations to ensure naturally aligned data.
- Start with the largest size numeric items first, followed by smaller size numeric items, and then non-numeric (character) data.

The `!DEC$ ATTRIBUTES ALIGN` directive specifies the byte alignment for a variable. It is not supported for `ALLOCATABLE/POINTER` variables.

Common blocks (`COMMON` statement), derived-type data, and Fortran 77 record structures (`RECORD` statement) usually contain multiple items within the context of the larger structure.

The following statements can cause unaligned data:

Statement	Options	Description
Common blocks (COMMON statement)	`commons` or `dcommons`	The order of variables in the `COMMON` statement determines their storage order. Unless you are sure that the data items in the common block will be naturally aligned, specify either `-align commons` or `-align dcommons` (Linux*) or `/align:commons` or `/align:dcommons` (Windows*), depending on the largest data size used.


SDNL_2024_03_21 Property Pages
Configuration: Active(Release)
Platform: x64
Configuration Manager...
Configuration Properties
General
Debugging
Fortran
General
Optimization
Debugging
Preprocessor
Code Generation
Language
Compatibility
Diagnostics
Data
Floating Point
External Procedures
Output Files
Run-time
Libraries
Command Line
Linker
General
Input
Runtime Library
Multithreaded
Use Common Windows Libraries
No
Use Portlib Library
No
Use Intel Math Kernel Library
No
Use ILP64 interfaces
No
Disable Default Library Search Rules
No
Disable OBJCOMMENT Library Names in
No
Runtime Library
Specifies the runtime library for linking. (/libs:{static|dll|qwin|qwins}, /threads, /dbglibs)
OK
Cancel
Apply

SDNL_2024_03_21 Property Pages

Configuration: Active(Release) Platform: x64 Configuration Manager...

- Configuration Properties
  - General
  - Debugging
  - Fortran
    - General
    - Optimization
    - Debugging
    - Preprocessor
    - Code Generation
    - Language
    - Compatibility
    - Diagnostics
    - Data
    - Floating Point
    - External Procedures
    - Output Files
    - Run-time
    - Libraries
    - Command Line
  - Linker
  - Resources
  - MIDL

All Options:

```
/nologo /O2 /module:"x64\Release\\" /object:"x64\Release\\" /Fd"x64\Release\vc170.pdb"
/libs:static /threads /c
```

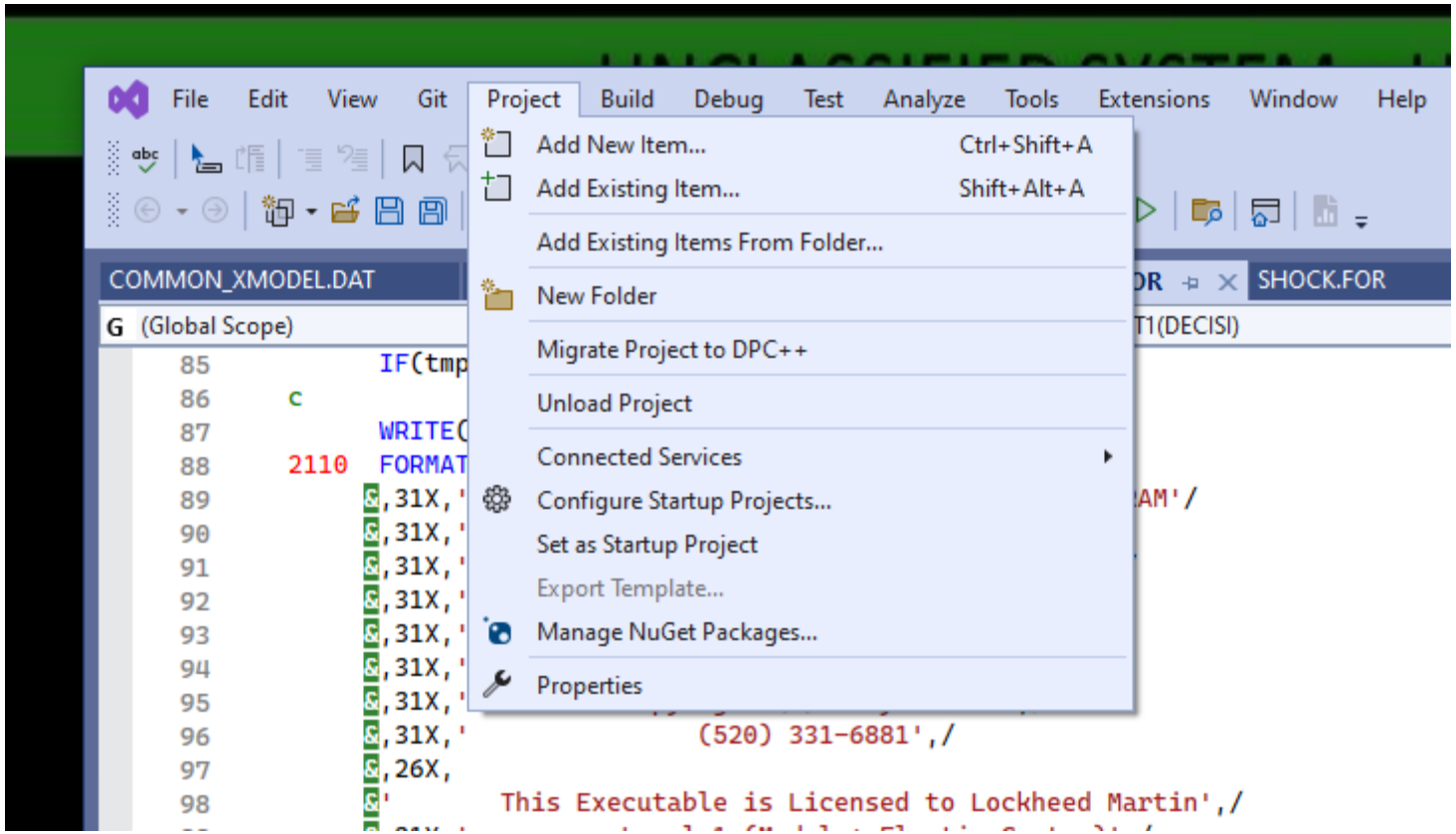
Additional Options:

```
/Qsave /align:dcommons
```

OK Cancel Apply

Fortran Compiler Option

Select Properties



Select the compiler to use
Select the item to get the dropdown box!!