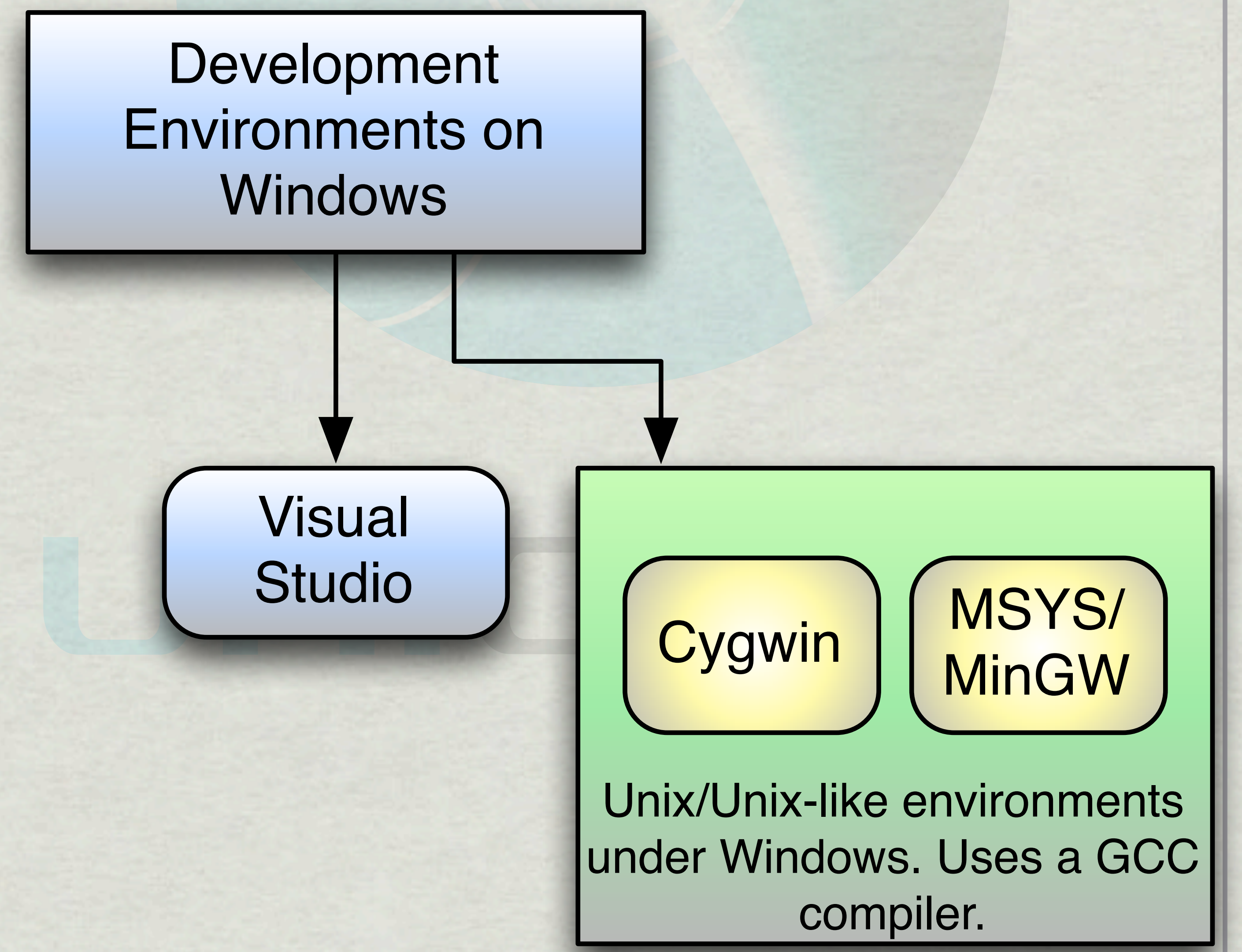
# NetCDF-C for Windows

NetCDF for New Users

2012

## Overview

- Focus of this discussion is on building and using netCDF-C on Windows.
- Multiple Windows development environments:
  - Cygwin
  - MSYS/MinGW
  - Visual Studio
- We will focus on using CMake to build Visual-Studio compatible netCDF-C libraries.


Development Environments on Windows
Visual Studio
Cygwin
MSYS/ MinGW
Unix/Unix-like environments under Windows. Uses a GCC compiler.

# Getting netCDF-C

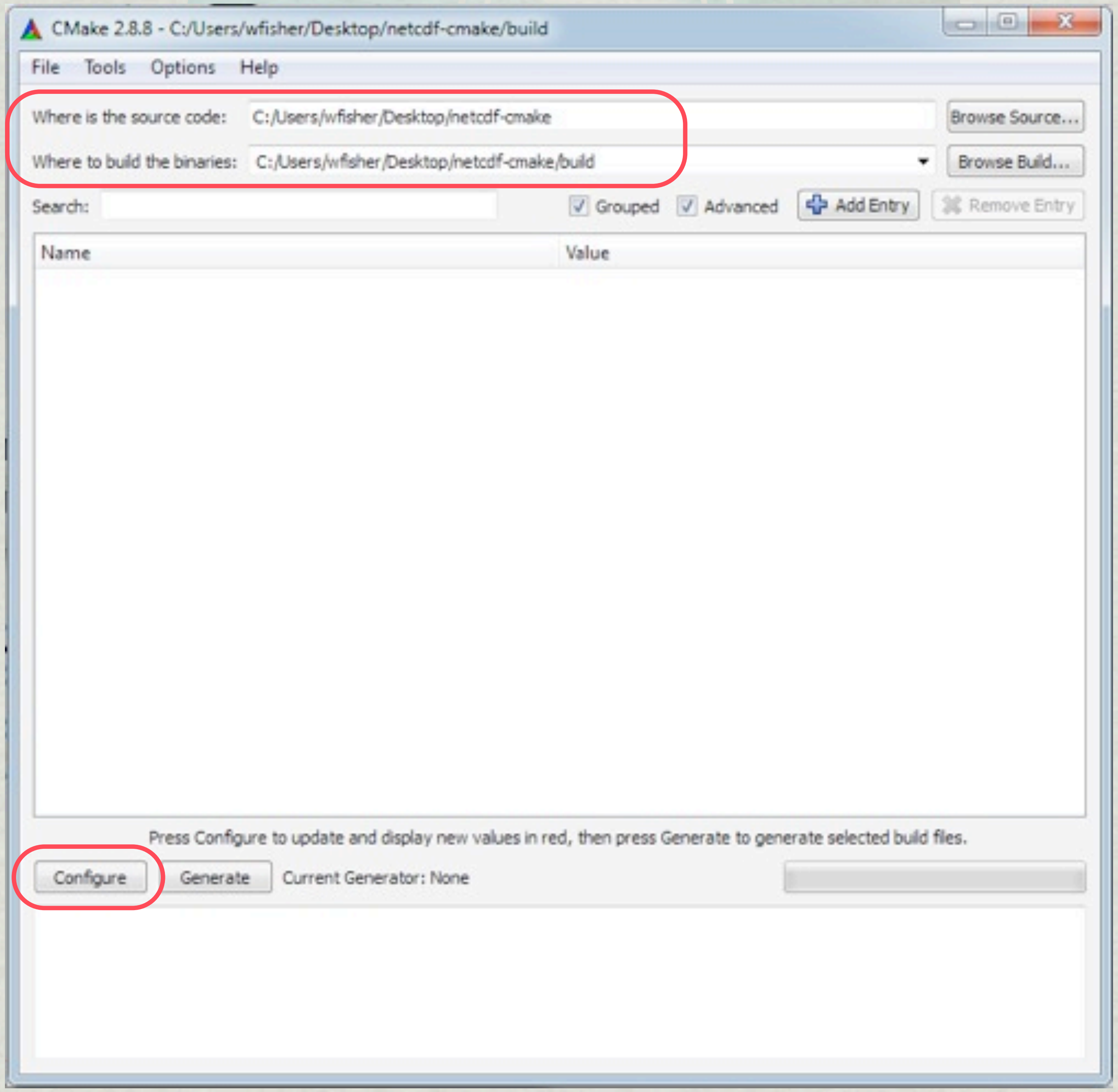
- Latest Stable release (4.2.1.1):
  - http://www.unidata.ucar.edu/downloads/netcdf
- Latest Developer Snapshot:
  - svn co http://svn.unidata.ucar.edu/repos/netcdf/trunk

## Getting CMake

- http://www.cmake.org
- Current version is 2.8.9.
    - netCDF-C requires at least CMake 2.8.8
- The CMake download comes with both command line and GUI tools.

# Using the CMake GUI

- The source directory and the build directory are typically two distinct locations.
- Step 1) Set locations.
- Step 2) Configure


CMake 2.8.8 - C:/Users/wfisher/Desktop/netcdf-cmake/build

Where is the source code: C:/Users/wfisher/Desktop/netcdf-cmake

Where to build the binaries: C:/Users/wfisher/Desktop/netcdf-cmake/build

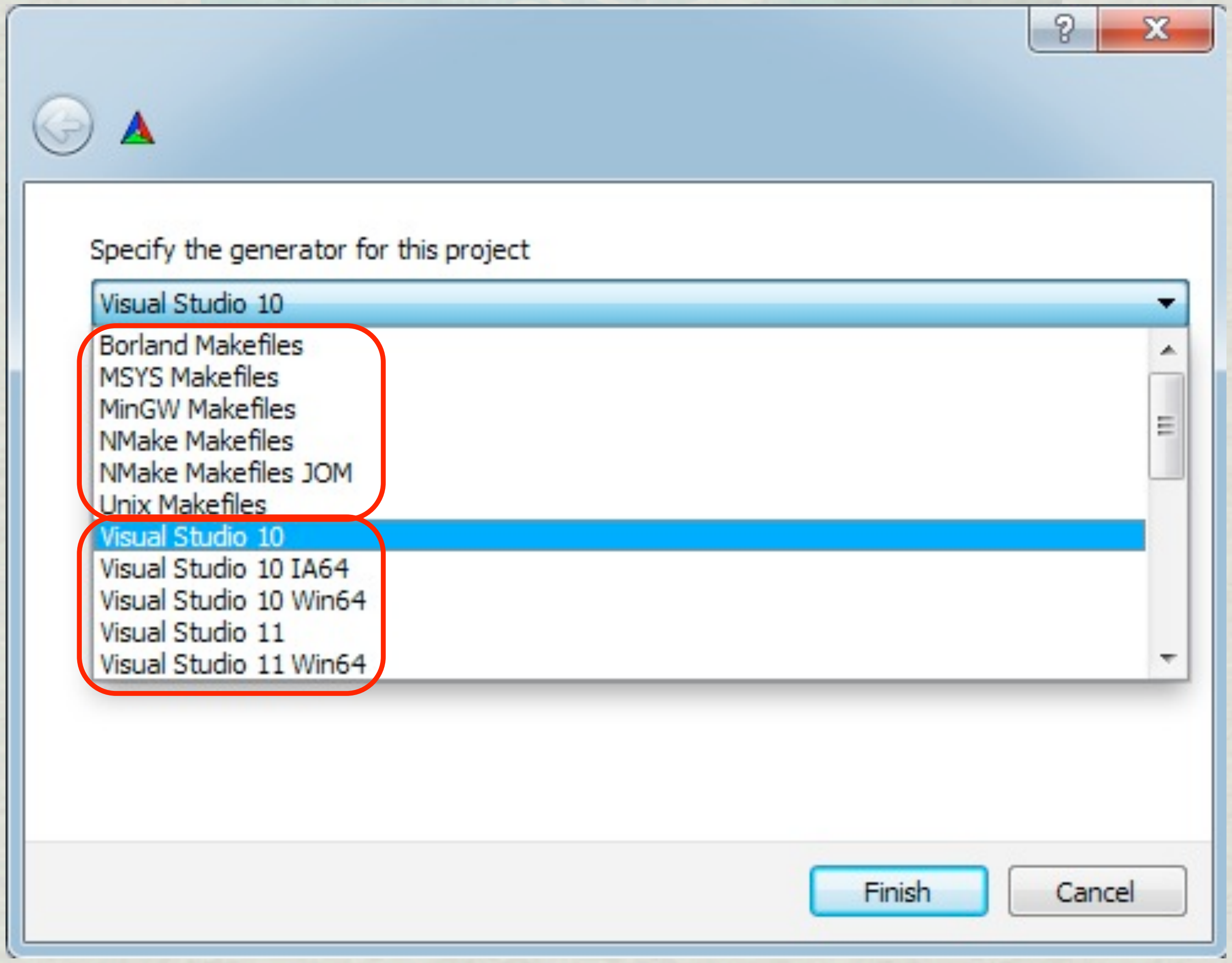
Search: ☑ Grouped ☑ Advanced

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None

# Using the CMake GUI

- CMake will ask you to specify which ‘Generator’ to use.
  - Different makefile-based builds.
  - Visual Studio builds, specific to desired architecture.



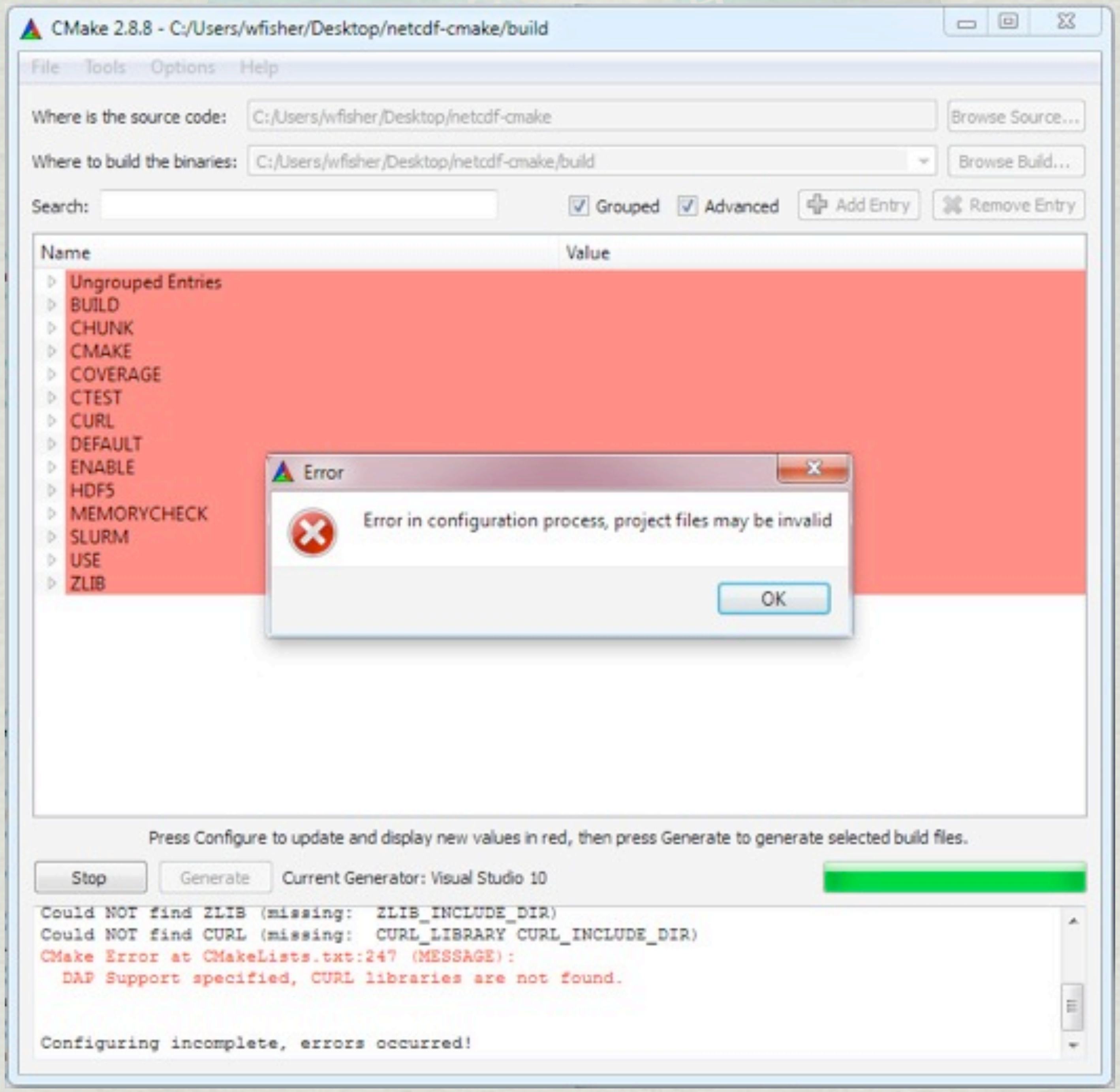
Specify the generator for this project

Visual Studio 10

```
Borland Makefiles
MSYS Makefiles
MinGW Makefiles
NMake Makefiles
NMake Makefiles JOM
Unix Makefiles
Visual Studio 10
Visual Studio 10 IA64
Visual Studio 10 Win64
Visual Studio 11
Visual Studio 11 Win64
```

# Using the CMake GUI

- Here we see an error; DAP is enabled by default, but the required curl libraries couldn't be located.



CMake 2.8.8 - C:/Users/wfisher/Desktop/netcdf-cmake/build

Where is the source code: C:/Users/wfisher/Desktop/netcdf-cmake

Where to build the binaries: C:/Users/wfisher/Desktop/netcdf-cmake/build

Search:

☑ Grouped ☑ Advanced

| Name | Value |
|---|---|
| Ungrouped Entries | |
| BUILD | |
| CHUNK | |
| CMAKE | |
| COVERAGE | |
| CTEST | |
| CURL | |
| DEFAULT | |
| ENABLE | |
| HDF5 | |
| MEMORYCHECK | |
| SLURM | |
| USE | |
| ZLIB | |

Error

Error in configuration process, project files may be invalid

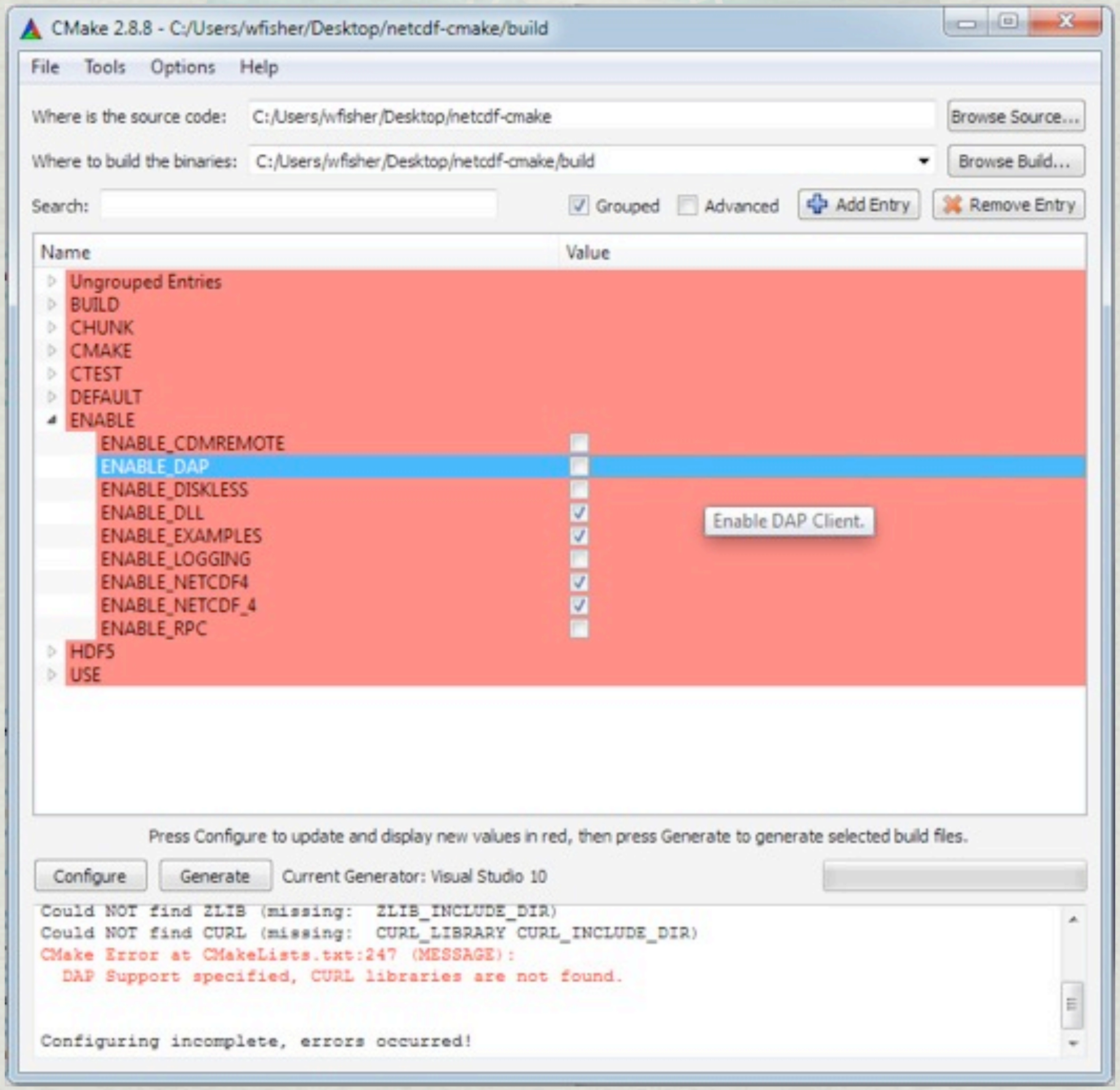
Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 10

```
Could NOT find ZLIB (missing:  ZLIB_INCLUDE_DIR)
Could NOT find CURL (missing:  CURL_LIBRARY CURL_INCLUDE_DIR)
CMake Error at CMakeLists.txt:247 (MESSAGE):
  DAP Support specified, CURL libraries are not found.


Configuring incomplete, errors occurred!
```

# Using the CMake GUI

- Under ‘enable’ options, we uncheck ‘ENABLE_DAP’, then re-run ‘configure’.



Where is the source code: C:/Users/wfisher/Desktop/netcdf-cmake

Where to build the binaries: C:/Users/wfisher/Desktop/netcdf-cmake/build

Search:

[✓ Grouped] [ ] Advanced

| Name | Value |
|---|---|
| Ungrouped Entries | |
| BUILD | |
| CHUNK | |
| CMAKE | |
| CTEST | |
| DEFAULT | |
| ENABLE | |
| ENABLE_CDMREMOTE | ☐ |
| ENABLE_DAP | ☐ |
| ENABLE_DISKLESS | ☐ |
| ENABLE_DLL | ✓ |
| ENABLE_EXAMPLES | ✓ |
| ENABLE_LOGGING | ☐ |
| ENABLE_NETCDF4 | ✓ |
| ENABLE_NETCDF_4 | ✓ |
| ENABLE_RPC | ☐ |
| HDF5 | |
| USE | |

Enable DAP Client.

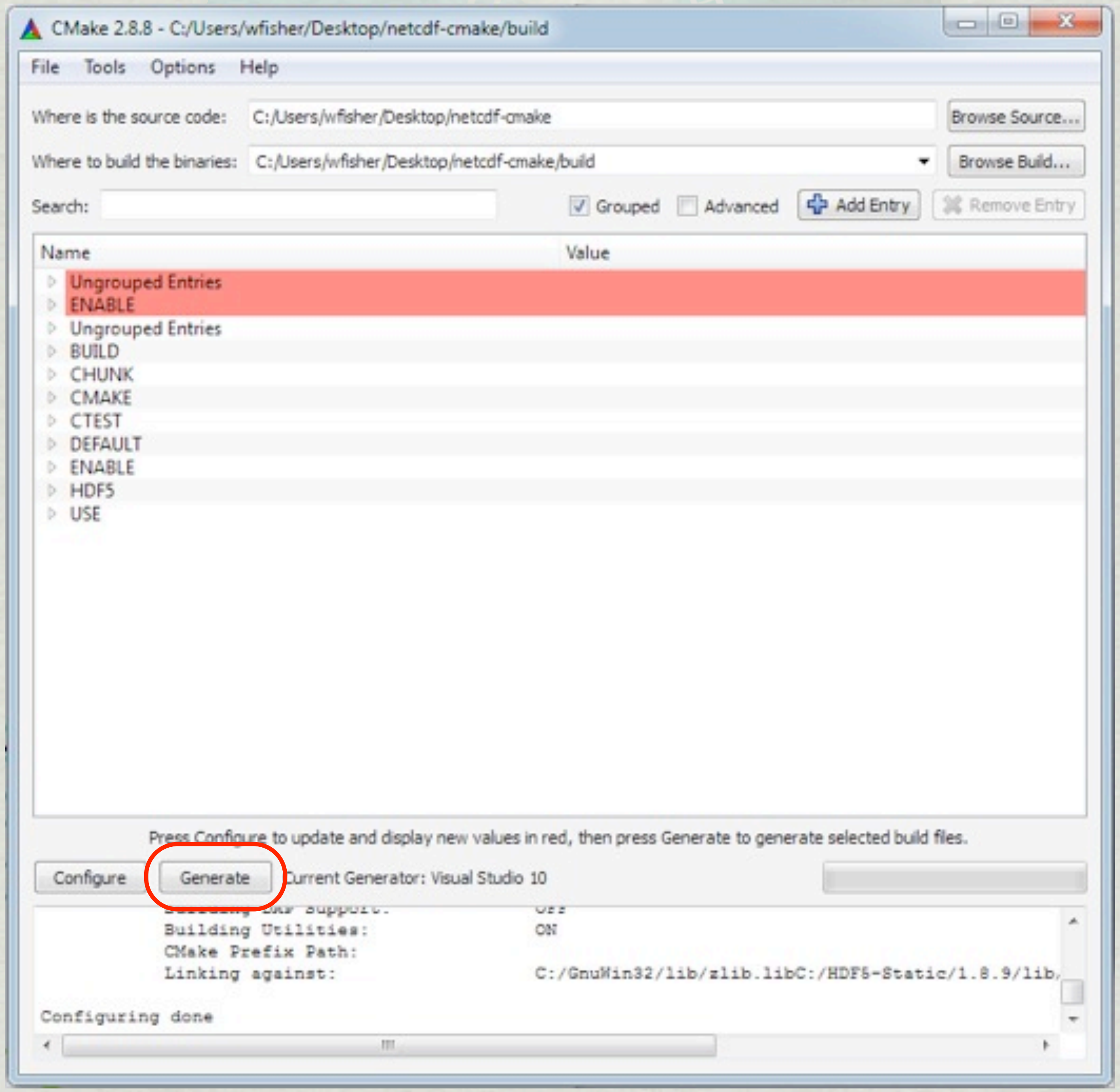
Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 10

```
Could NOT find ZLIB (missing:  ZLIB_INCLUDE_DIR)
Could NOT find CURL (missing:  CURL_LIBRARY CURL_INCLUDE_DIR)
CMake Error at CMakeLists.txt:247 (MESSAGE):
  DAP Support specified, CURL libraries are not found.


Configuring incomplete, errors occurred!
```

# Using the CMake GUI

- Configuration was successful.
- Next, we generate the project files.



Where is the source code: C:/Users/wfisher/Desktop/netcdf-cmake

Where to build the binaries: C:/Users/wfisher/Desktop/netcdf-cmake/build

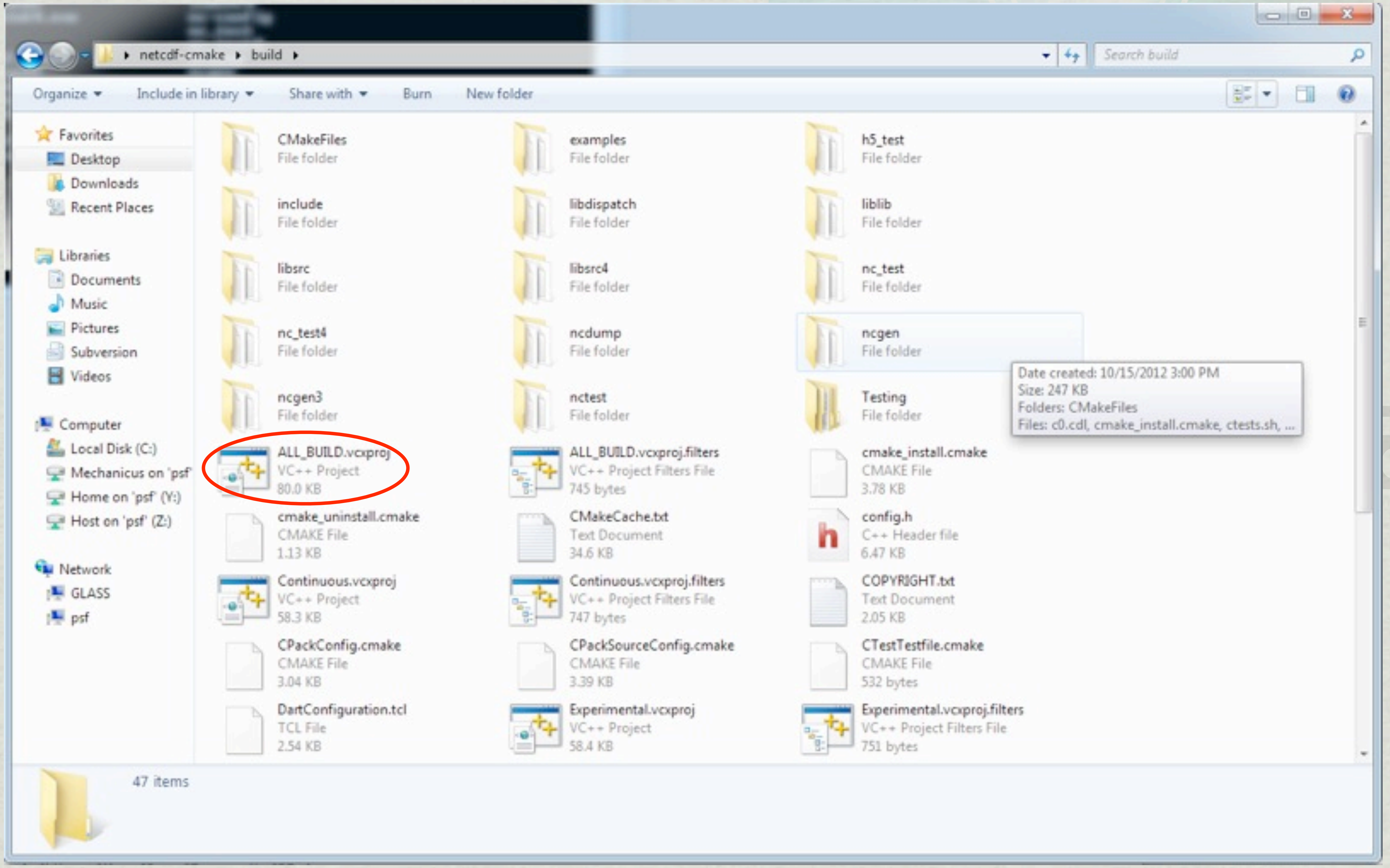
| Name | Value |
|---|---|
| Ungrouped Entries | |
| ENABLE | |
| Ungrouped Entries | |
| BUILD | |
| CHUNK | |
| CMAKE | |
| CTEST | |
| DEFAULT | |
| ENABLE | |
| HDF5 | |
| USE | |

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 10

```
        Building Utilities:                ON
        CMake Prefix Path:
        Linking against:                   C:/GnuWin32/lib/zlib.libC:/HDF5-Static/1.8.9/lib,

Configuring done
```

# Compiling CMake-generated Visual Studio Projects



CMakeFiles
File folder

examples
File folder

h5_test
File folder

include
File folder

libdispatch
File folder

liblib
File folder

libsrc
File folder

libsrc4
File folder

nc_test
File folder

nc_test4
File folder

ncdump
File folder

ncgen
File folder

ncgen3
File folder

nctest
File folder

Testing
File folder

Date created: 10/15/2012 3:00 PM
Size: 247 KB
Folders: CMakeFiles
Files: c0.cdl, cmake_install.cmake, ctests.sh, ...

ALL_BUILD.vcxproj
VC++ Project
80.0 KB

ALL_BUILD.vcxproj.filters
VC++ Project Filters File
745 bytes

cmake_install.cmake
CMAKE File
3.78 KB

cmake_uninstall.cmake
CMAKE File
1.13 KB

CMakeCache.txt
Text Document
34.6 KB

config.h
C++ Header file
6.47 KB

Continuous.vcxproj
VC++ Project
58.3 KB

Continuous.vcxproj.filters
VC++ Project Filters File
747 bytes

COPYRIGHT.txt
Text Document
2.05 KB

CPackConfig.cmake
CMAKE File
3.04 KB

CPackSourceConfig.cmake
CMAKE File
3.39 KB

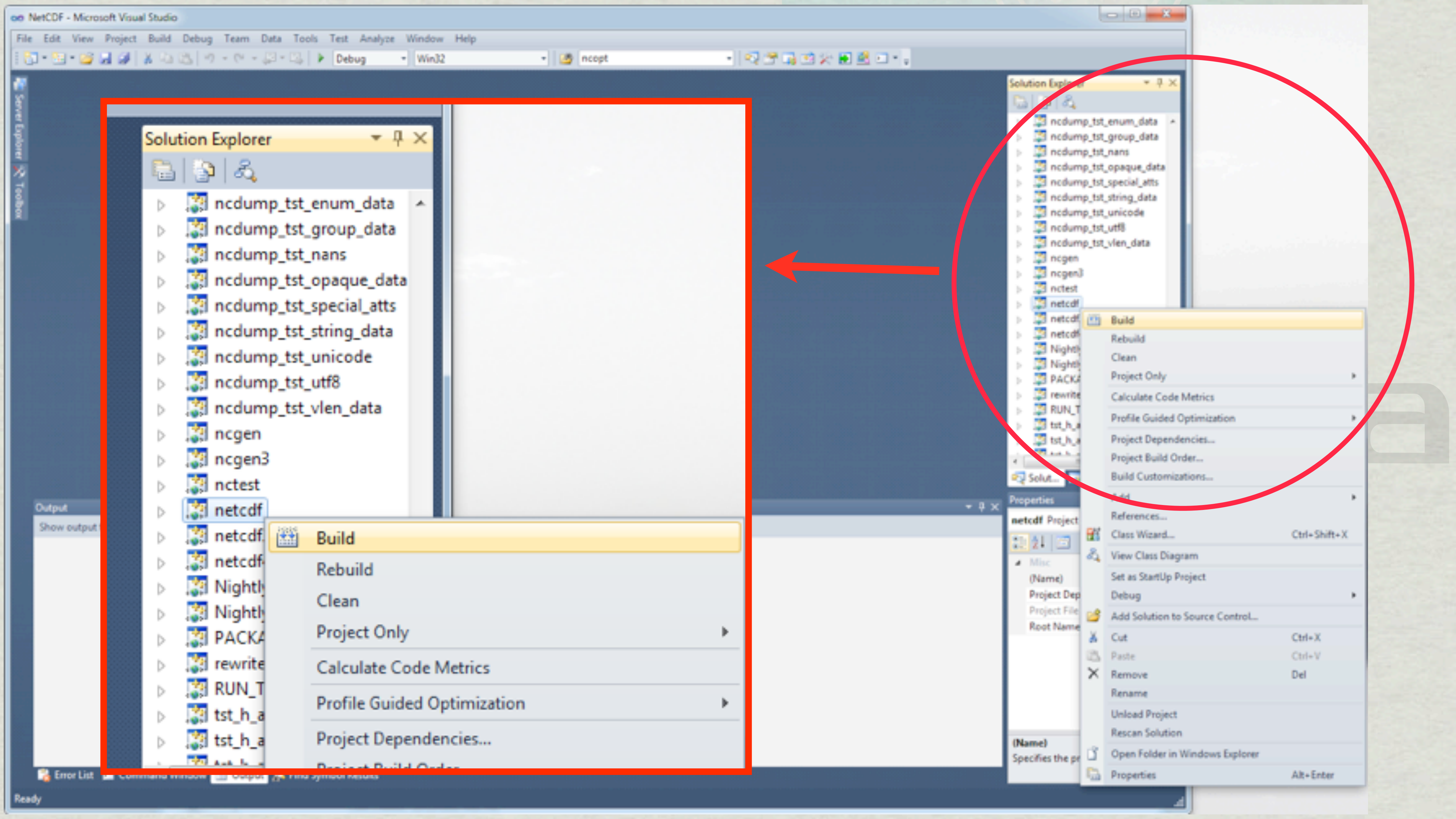
CTestTestfile.cmake
CMAKE File
532 bytes

DartConfiguration.tcl
TCL File
2.54 KB

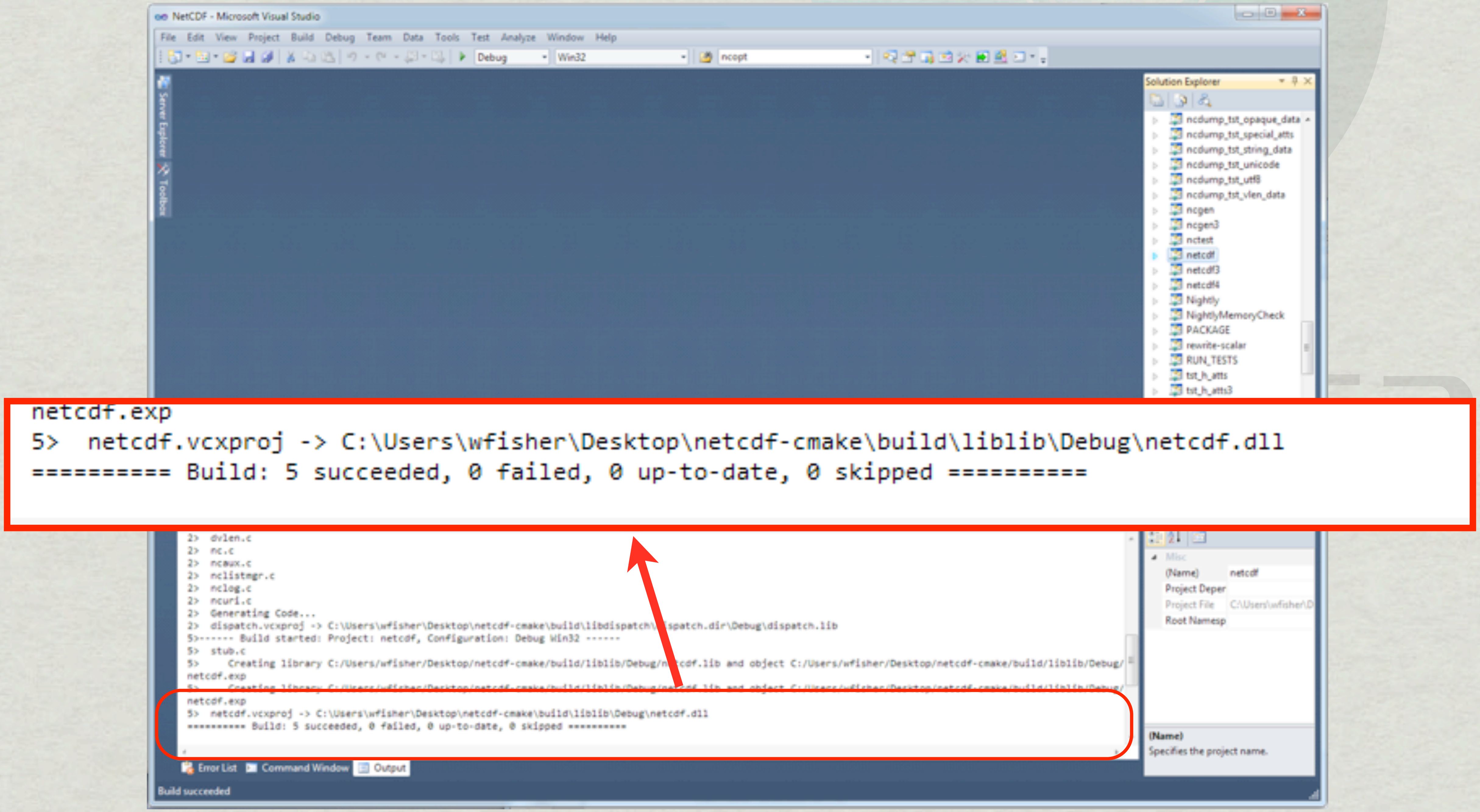
Experimental.vcxproj
VC++ Project
58.4 KB

Experimental.vcxproj.filters
VC++ Project Filters File
751 bytes

# Compiling CMake-generated Visual Studio Projects



# Compiling CMake-generated Visual Studio Projects

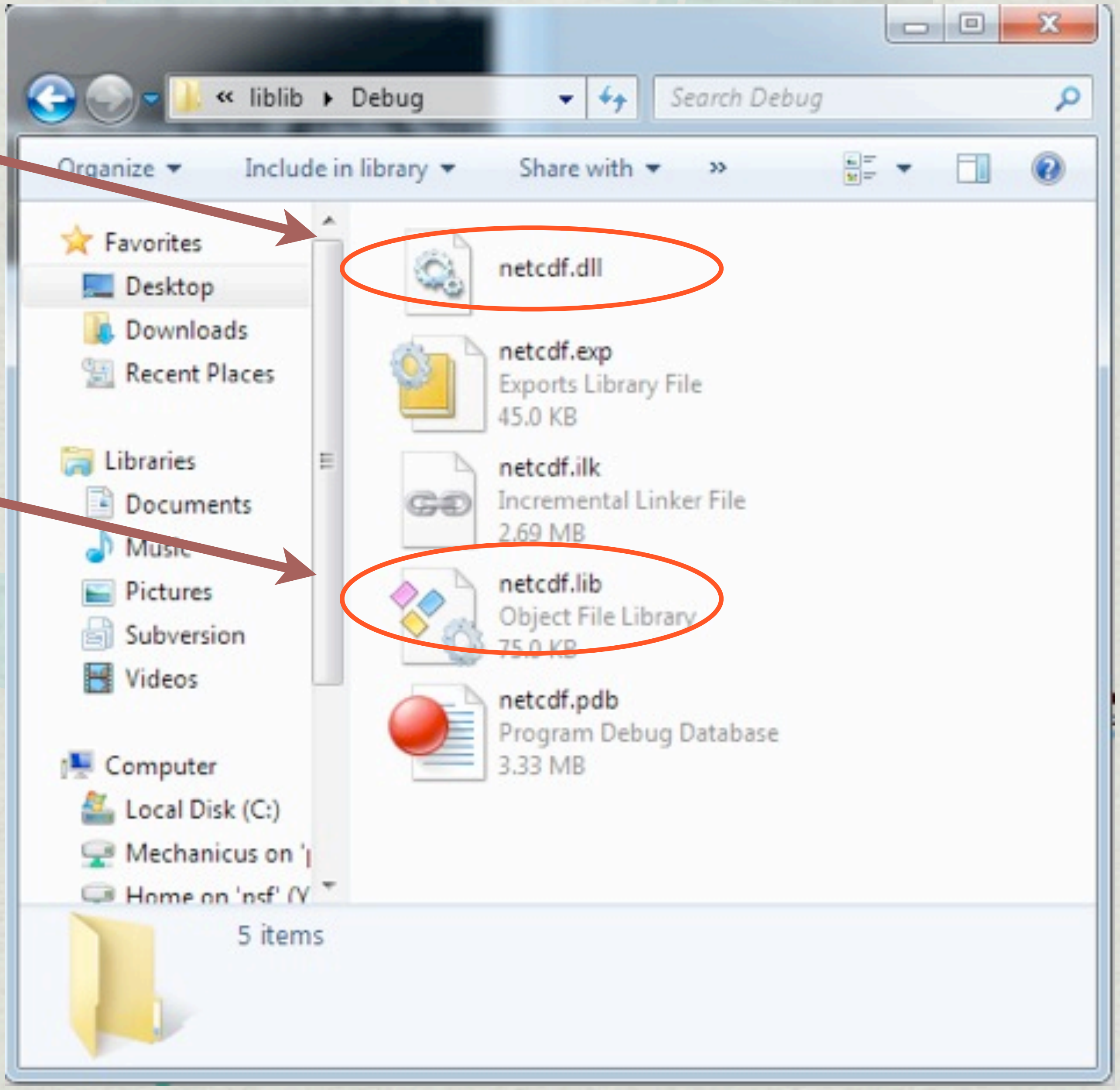


```
netcdf.exp
5>  netcdf.vcxproj -> C:\Users\wfisher\Desktop\netcdf-cmake\build\liblib\Debug\netcdf.dll
========== Build: 5 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

```
2>  dvlen.c
2>  nc.c
2>  ncaux.c
2>  nclistmgr.c
2>  nclog.c
2>  ncuri.c
2>  Generating Code...
2>  dispatch.vcxproj -> C:\Users\wfisher\Desktop\netcdf-cmake\build\libdispatch\dispatch.dir\Debug\dispatch.lib
5>------ Build started: Project: netcdf, Configuration: Debug Win32 ------
5>  stub.c
5>     Creating library C:/Users/wfisher/Desktop/netcdf-cmake/build/liblib/Debug/netcdf.lib and object C:/Users/wfisher/Desktop/netcdf-cmake/build/liblib/Debug/
netcdf.exp
5>     Creating library C:/Users/wfisher/Desktop/netcdf-cmake/build/liblib/Debug/netcdf.lib and object C:/Users/wfisher/Desktop/netcdf-cmake/build/liblib/Debug/
netcdf.exp
5>  netcdf.vcxproj -> C:\Users\wfisher\Desktop\netcdf-cmake\build\liblib\Debug\netcdf.dll
========== Build: 5 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

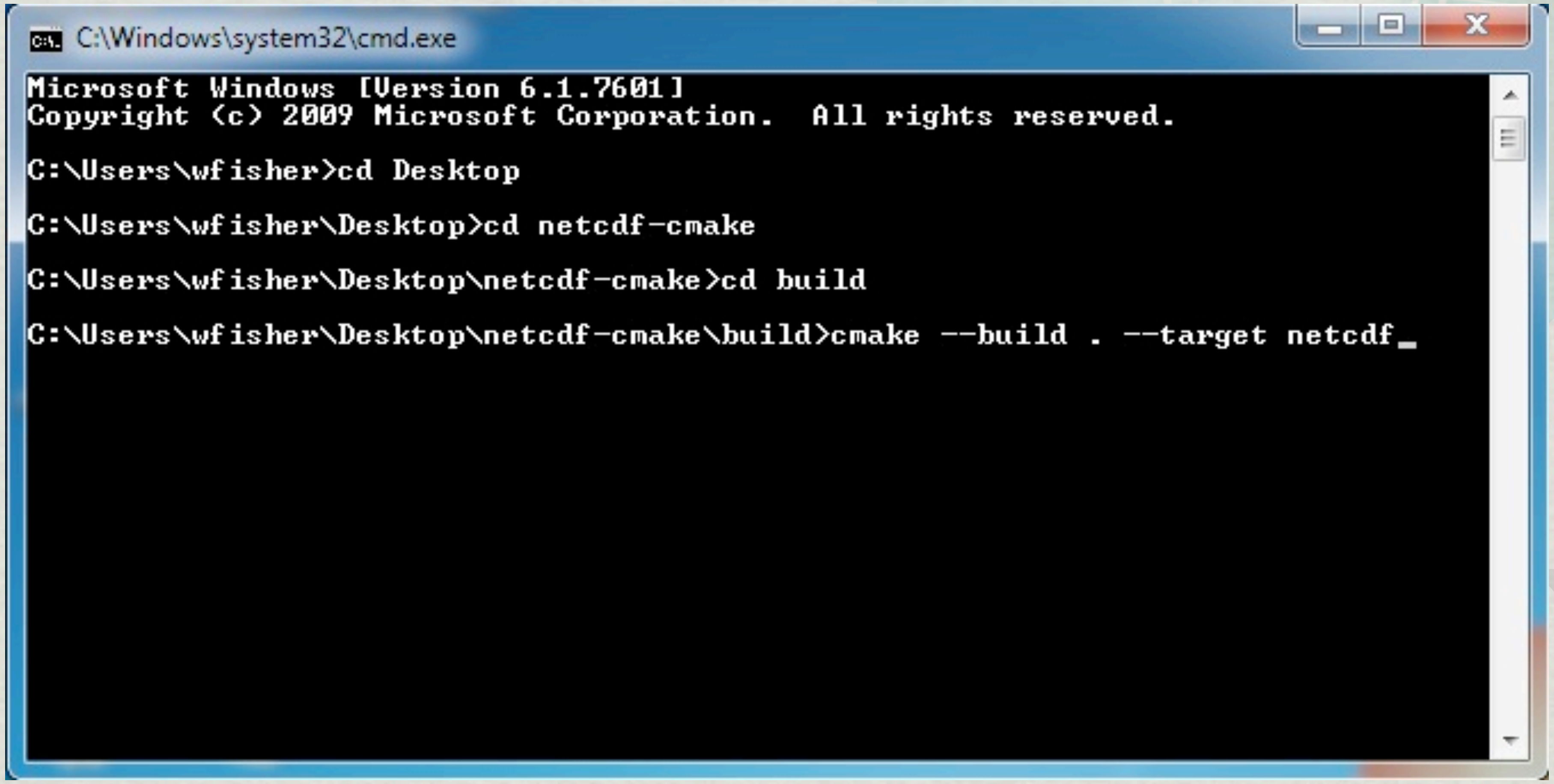
# Compiling CMake-generated Visual Studio Projects

- On Windows, shared libraries are not linked against directly.
- 'Import Libraries' are instead used.
- What if we wanted to do this from the command line?



netcdf.dll
netcdf.exp
Exports Library File
45.0 KB
netcdf.ilk
Incremental Linker File
2.69 MB
netcdf.lib
Object File Library
75.0 KB
netcdf.pdb
Program Debug Database
3.33 MB

## Building netCDF-C in Windows via the Command Line

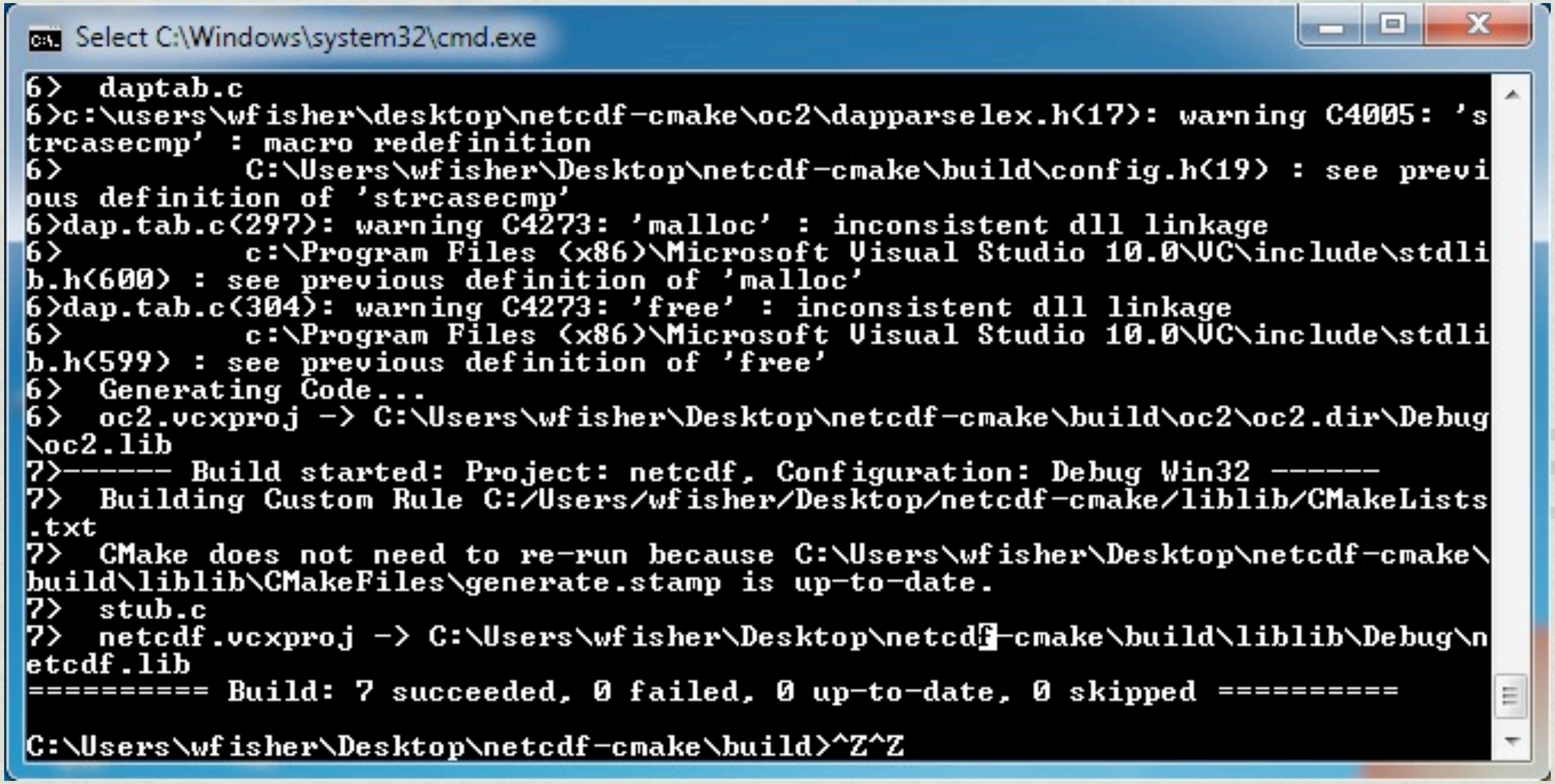

C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\wfisher>cd Desktop

C:\Users\wfisher\Desktop>cd netcdf-cmake

C:\Users\wfisher\Desktop\netcdf-cmake>cd build

C:\Users\wfisher\Desktop\netcdf-cmake\build>cmake --build . --target netcdf_
```

```
C:\netcdf> cmake --build . --target netcdf
```

```
C:\netcdf> cmake --build .
```

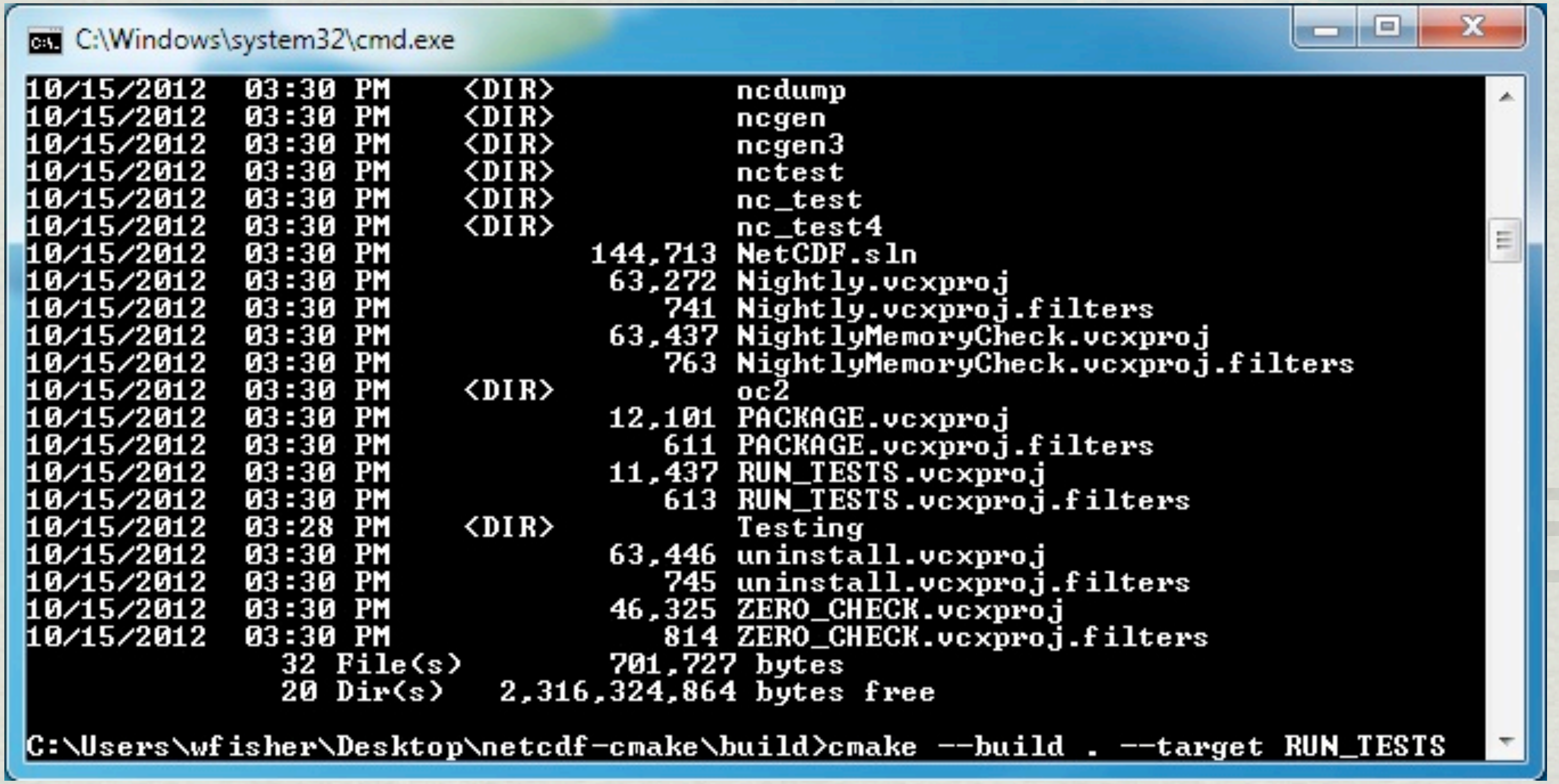
## Building netCDF-C in Windows via the Command Line



Select C:\Windows\system32\cmd.exe

```
6>  daptab.c
6>c:\users\wfisher\desktop\netcdf-cmake\oc2\dapparselex.h(17): warning C4005: 's
trcasecmp' : macro redefinition
6>          C:\Users\wfisher\Desktop\netcdf-cmake\build\config.h(19) : see previ
ous definition of 'strcasecmp'
6>dap.tab.c(297): warning C4273: 'malloc' : inconsistent dll linkage
6>          c:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\include\stdli
b.h(600) : see previous definition of 'malloc'
6>dap.tab.c(304): warning C4273: 'free' : inconsistent dll linkage
6>          c:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\include\stdli
b.h(599) : see previous definition of 'free'
6>  Generating Code...
6>  oc2.vcxproj -> C:\Users\wfisher\Desktop\netcdf-cmake\build\oc2\oc2.dir\Debug
\oc2.lib
7>------ Build started: Project: netcdf, Configuration: Debug Win32 ------
7>  Building Custom Rule C:/Users/wfisher/Desktop/netcdf-cmake/liblib/CMakeLists
.txt
7>  CMake does not need to re-run because C:\Users\wfisher\Desktop\netcdf-cmake\
build\liblib\CMakeFiles\generate.stamp is up-to-date.
7>  stub.c
7>  netcdf.vcxproj -> C:\Users\wfisher\Desktop\netcdf-cmake\build\liblib\Debug\n
etcdf.lib
========== Build: 7 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========

C:\Users\wfisher\Desktop\netcdf-cmake\build>^Z^Z
```

# Running NetCDF Tests in Windows

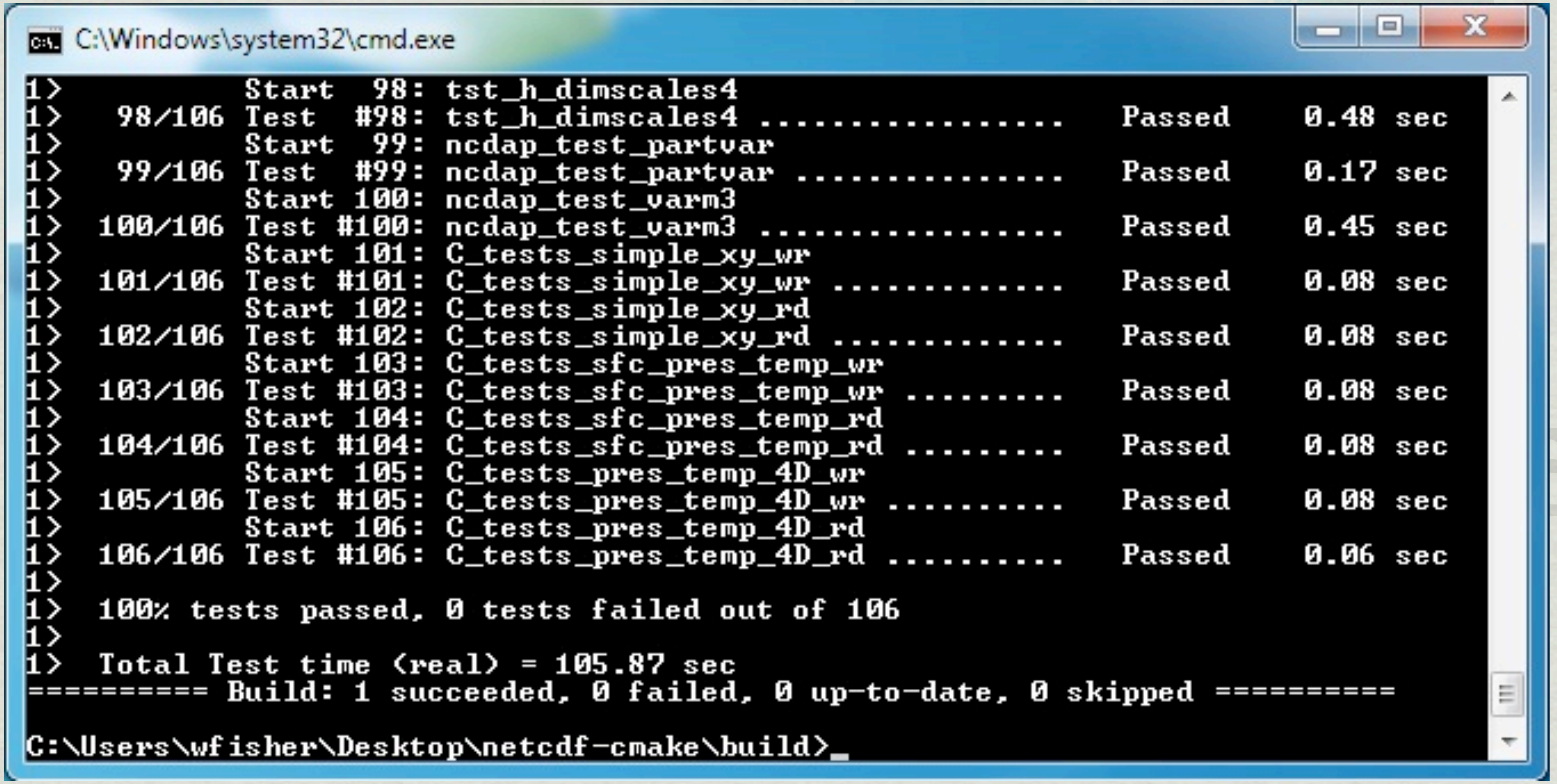


```
10/15/2012  03:30 PM    <DIR>          ncdump
10/15/2012  03:30 PM    <DIR>          ncgen
10/15/2012  03:30 PM    <DIR>          ncgen3
10/15/2012  03:30 PM    <DIR>          nctest
10/15/2012  03:30 PM    <DIR>          nc_test
10/15/2012  03:30 PM    <DIR>          nc_test4
10/15/2012  03:30 PM           144,713 NetCDF.sln
10/15/2012  03:30 PM            63,272 Nightly.vcxproj
10/15/2012  03:30 PM               741 Nightly.vcxproj.filters
10/15/2012  03:30 PM            63,437 NightlyMemoryCheck.vcxproj
10/15/2012  03:30 PM               763 NightlyMemoryCheck.vcxproj.filters
10/15/2012  03:30 PM    <DIR>          oc2
10/15/2012  03:30 PM            12,101 PACKAGE.vcxproj
10/15/2012  03:30 PM               611 PACKAGE.vcxproj.filters
10/15/2012  03:30 PM            11,437 RUN_TESTS.vcxproj
10/15/2012  03:30 PM               613 RUN_TESTS.vcxproj.filters
10/15/2012  03:28 PM    <DIR>          Testing
10/15/2012  03:30 PM            63,446 uninstall.vcxproj
10/15/2012  03:30 PM               745 uninstall.vcxproj.filters
10/15/2012  03:30 PM            46,325 ZERO_CHECK.vcxproj
10/15/2012  03:30 PM               814 ZERO_CHECK.vcxproj.filters
              32 File(s)        701,727 bytes
              20 Dir(s)   2,316,324,864 bytes free

C:\Users\wfisher\Desktop\netcdf-cmake\build>cmake --build . --target RUN_TESTS
```

```
C:\netcdf> cmake --build . --target RUN_TESTS
```

```
C:\netcdf> ctest .
```

# Running NetCDF Tests in Windows



```
1>             Start  98: tst_h_dimscales4
1>    98/106 Test  #98: tst_h_dimscales4 .................   Passed    0.48 sec
1>             Start  99: ncdap_test_partvar
1>    99/106 Test  #99: ncdap_test_partvar ...............   Passed    0.17 sec
1>             Start 100: ncdap_test_varm3
1>   100/106 Test #100: ncdap_test_varm3 .................   Passed    0.45 sec
1>             Start 101: C_tests_simple_xy_wr
1>   101/106 Test #101: C_tests_simple_xy_wr .............   Passed    0.08 sec
1>             Start 102: C_tests_simple_xy_rd
1>   102/106 Test #102: C_tests_simple_xy_rd .............   Passed    0.08 sec
1>             Start 103: C_tests_sfc_pres_temp_wr
1>   103/106 Test #103: C_tests_sfc_pres_temp_wr .........   Passed    0.08 sec
1>             Start 104: C_tests_sfc_pres_temp_rd
1>   104/106 Test #104: C_tests_sfc_pres_temp_rd .........   Passed    0.08 sec
1>             Start 105: C_tests_pres_temp_4D_wr
1>   105/106 Test #105: C_tests_pres_temp_4D_wr ..........   Passed    0.08 sec
1>             Start 106: C_tests_pres_temp_4D_rd
1>   106/106 Test #106: C_tests_pres_temp_4D_rd ..........   Passed    0.06 sec
1>
1>   100% tests passed, 0 tests failed out of 106
1>
1>   Total Test time (real) = 105.87 sec
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========

C:\Users\wfisher\Desktop\netcdf-cmake\build>_
```

## Summary

- We have discussed
  - The motivation for including CMake support in netCDF-C.
  - Configuring, building and testing netCDF-C on Windows using CMake & Visual Studio.
    - GUI
    - Command Line