WATFLOOD® / CHARM®

Canadian Hydrological And Routing Model

Programmers Manual

SINCE 1972

Developed for

Surveys and Information Branch
Ecosystem Science and Evaluation Directorate
ENVIRONMENT CANADA

by

Nicholas Kouwen, Ph.D., P.Eng., F.ASCE
Distinguished Professor Emeritus
Department of Civil Engineering
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1
519-922-2602

E-mail:
kouwen@uwaterloo.ca

http://www.watflood.ca

First Edition: December 18, 2018

1 WATFLOOD®/CHARM® Programmers Manual

WATFLOOD is a registered trademark
CHARM is the acronym for Canadian Hydrological And Routing Model – registered trademark pending

Green Kenue and WATFLOOD were developed for and financially supported by
Surveys and Information Branch
Ecosystem Science and Evaluation Directorate
ENVIRONMENT CANADA

Contributors:

Tricia Stadnyk	wetland routing, tracers and isotope model
Tegan Holmes	tracers and isotope model
Todd Neff	evapo-transpiration model
John Donald	snow model
Frank Seglinieks	snow model
Ric Soulis	advisor to Todd Neff, John Donald & Frank Seglenieks

1.1 WATFLOOD overview

WATFLOOD has three distinct components:
CHARM*.exe – the hydrological and routing model
SNW*.exe, MOIST*.exe, RAGMET*.exe and TMP*.exe – point data to gridded data conversion
And then a host of data processing and conversion programs to ingest meteorological and hydrometric data.
The next page shows the complete list with links to the executables.
Following the list of executables are screenshots of the Solution Explorer in Microsoft Visual Studio showing the subroutines (s/r's) in each program

1.2 Downloading WATFLOOD code

The watflood code is released as open source with the GNU Lesser General Public License
http://www.gnu.org/licenses/
The code may be downloaded by installing TortoiseSVN with the following folders on your computer:
Watflood\model
Watflood\utilities
Watflood\common

The setting the repository browser for each folder:
https://chyms.nrc.gc.ca/svn/watflood/common
https://chyms.nrc.gc.ca/svn/watflood/iso
https://chyms.nrc.gc.ca/svn/watflood/model
https://chyms.nrc.gc.ca/svn/watflood/utilities

You should then be able to checkout the code

1.3 Conditions

As a condition to accessing, modifying, using and distributing the WATFLOOD code you agree NOT to remove any license information and/or disclaimers in the WATFLOOD code – e.g.:

```
!***********************************************************************
!    Copyright (C) 1987-2018 by Nicholas Kouwen

!    This file is part of WATFLOOD (R)

!    WATFLOOD(R) is free software: you can redistribute it and/or modify
!    it under the terms of the GNU Lesser General Public License as published by
!    the Free Software Foundation, either version 3 of the License, or
!    any later version.

!    WATFLOOD is distributed in the hope that it will be useful,
!    but WITHOUT ANY WARRANTY; without even the implied warranty of
!    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
!    GNU Lesser General Public License for more details.

!    You should have received a copy of the GNU Lesser General Public License
!    along with WATFLOOD.  If not, see <http://www.gnu.org/licenses/>.
!***********************************************************************
```

```
      print*,'DISCLAIMER'
       print*,'The WATFLOOD software and other material supplied'
       print*,'in connection herewith is furnished by N. Kouwen and the'
       print*,'University of Waterloo and is accepted by the'
       print*,'user upon the express understanding that N. Kouwen'
       print*,'or the University of Waterloo make no warranties, either'
       print*,'express or implied, concerning the accuracy, completeness,'
       print*,'reliability, usability, performance, or fitness for any'
       print*,'particular purpose.'
       print*
       print*,'The material is provided "as is". The entire risk as to'
       print*,'its quality and performance is with the user.'
      print*
       print*,'The forecasts produced by the WATFLOOD software are for'
       print*,'information and discussion purposes only and are not to'
       print*,'be relied upon in any particular situation without the'
       print*,'express written consent of N. Kouwen or the'
       print*,'University of Waterloo.'
      print*
```

1.4 WATFLOOD Components

Function	Executable
Watershed model	
CHARM64x & CHARM64d Download the model executables by agreeing to accept the Disclaimers in the WATFLOOD manual and those stated in the code.	To get CHARM x & d respectively: I accept & I accept the disclaimer
Utilities (WATFLOOD Manual)	manual
.map to _shd.r2c converter	BSN64X.ZIP BSN64D.ZIP
Snow course distribution	SNW64.ZIP
Soil Moisture distribution	moist64.zip
Precipitation distribution	ragmet64x.zip ragmet64d.zip
Temperature distribution	tmp64x.zip tmp64d.zip
Daily Temperature Differences	diff.zip
Event file generator	make_evt.zip
Dynamically Dimensioned Search - DDS http://www.civil.uwaterloo.ca/btolson/papers.htm	DDS_p.zip
WATFLOOD-DDS coupler	dds_wfld_rev5.zip
Statistical analysis	stats64.zip
Model setup & Env. Can. data conversion	Utilities_Manual.pdf
Selecting CCC climate stations	select
Convert CCC climate data to tb0 format	ECmet
Convert WSC HYDAT flow data to tb0 format	ECflw
Convert WSC HYDAT flow data to tb0 format	ECrel
Convert WSC HYDAT level data to tb0 format	EClvl
Disaggredation of Gridded Precipitation NEW	blend64x.zip blend64d.zip
Forecasting	Flow_Forecasting_Manual.pdf
Create bat files for downloading CaPA, Regl & Glb forecasts & WSC provisional flows	run_daily
Convert CMC Regl. forecast to watershed r2c format	regl_conv.zip
Convert CMC glb. forecast to watershed r2c format	glb_conv.zip
Convert WSC downloaded provisional flows to tb0 format	wsc_rt.zip
File conversion - old to new formats	TRNS.ZIP
	FLI.ZIP (not supported)
	RTE.ZIP (not supported)

1.5 Renaming files

https://stackoverflow.com/questions/42842557/how-to-rename-file-on-svn-whilst-keeping-history

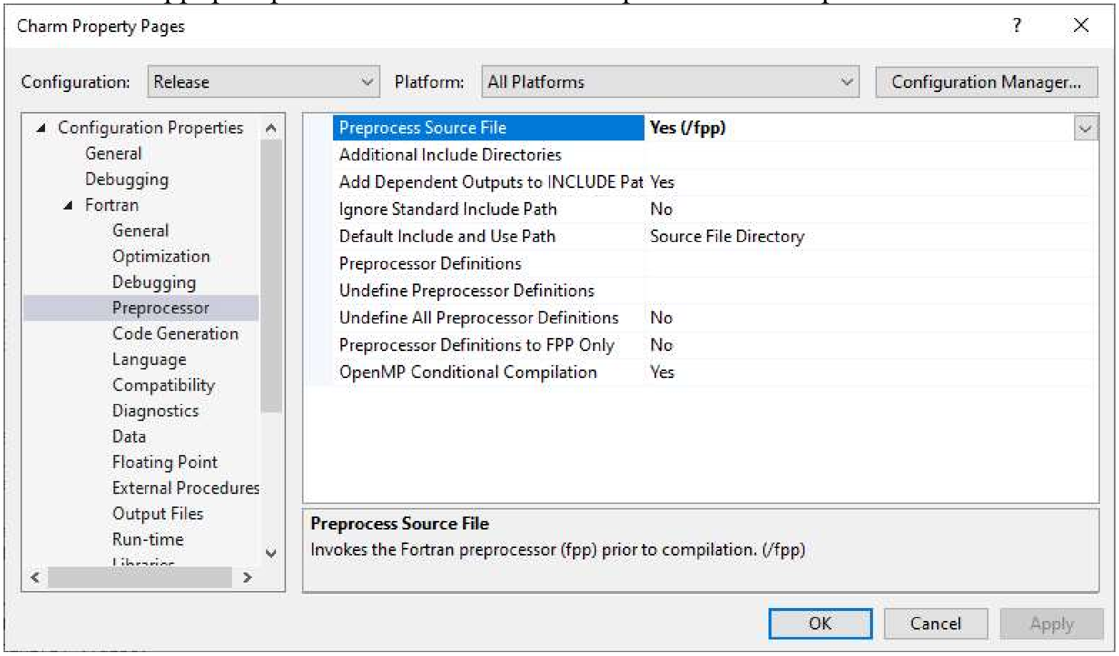
Since you have `tortoisesvn` tag in your question, this is how to do it using TortoiseSVN 1.9.7:

1. in your working copy folder right click on the file you want to rename and select "TortoiseSVN > Repo-browser"
2. **in the Repo-browser window** that opens, right click on the file you want to rename and select **"Copy to..."**
3. a dialog box appears asking you for the new file name (the file is shown with its full path), don't change the path, just change the file name and click OK.
4. still in the Repo-browser window, right click on the old file and select "Delete"

Done, the file was renamed and it kept all its history.

5. go back to your working copy folder, right click on the folder and select "TortoiseSVN > Check for Modification", then click "Check repository" button, right click on the renamed file and select "Update"

To use the /fpp pre=processor use Fortran > Preprocessor > Preprocess Source File = Yes



Configuration: Release

Platform: All Platforms

Preprocess Source File	Yes (/fpp)
Additional Include Directories	
Add Dependent Outputs to INCLUDE Pat	Yes
Ignore Standard Include Path	No
Default Include and Use Path	Source File Directory
Preprocessor Definitions	
Undefine Preprocessor Definitions	
Undefine All Preprocessor Definitions	No
Preprocessor Definitions to FPP Only	No
OpenMP Conditional Compilation	Yes

Preprocess Source File

Invokes the Fortran preprocessor (fpp) prior to compilation. (/fpp)

convert epoch time to yyyymmdd hhmmss

Quote:

> > The fstat function, implemented in some Fortran compilers, returns,
> > among other things, the time a file was last modified, in seconds after
> > midnight, 1970. Does anyone have Fortran code to convert this to a date
> > and time in yyyymmdd hhmmss format?

If your compilation environment supports the POSIX routines, (Intel, for one), you can simply call PXFLOCALTIME to do this:

call PXFLOCALTIME (isecnds, iatime, ierror)

where

isecnds - input integer with # of seconds since jan 1, 1970

iatime - output array returning the following:

1 - seconds (0 - 61, for leap seconds)

2 - minutes (0 - 59)

3 - hours (0 - 23)

4 - day of the month (1 - 31)

5 - month of the year (1 - 12)

6 - Gregorian year (e.g., 2006)

7 - Day of the week (0 = sunday)

8 - Day of the year (1 - 366)

9 - DST flag (0 = standard, 1 = DST)

ierror - Returns 0 if successful, EINVAL if not.

Note that by the POSIX Standard, some of the return values are '1-based' in Fortran, whereas the C 'localtime' counterpart returns '0-based' values. (Off the top of my head, day of the year is one of them.)

Hope this helps, Walt

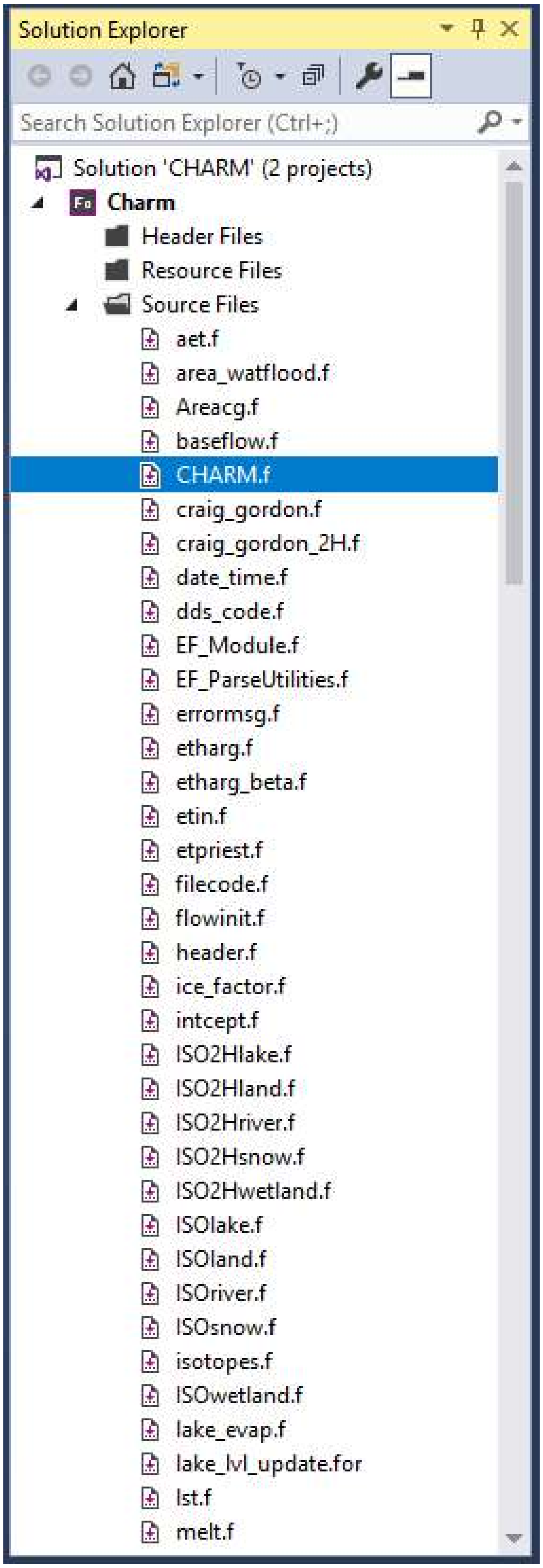
1.6 CHARM program components

CHARM is the hydrological model.
The first three charts show the subroutines making up CHARM
Two s/r's EF_Module.f and EF_Parse_utilities.f were written by Dave Watson at NRC as part of the effort to make all (or nearly all) WATFLOOD data files adhere to the Green Kenue formats. Some files were just not amenable to this – e.g. the event files and parameter files so these were left as text and Csv files. These modules are used by most WAFLOOD programs.
The module area_watflood.f is used by most WATFLOOD programs. Many of the utility programs are composed of the s/r's in the "common" folder.
The craig_gordon.f and iso*.f s/r's are for the isotope models
The remainder of the charts show the makeup of the utility programs.
The descriptions and users manuals for each of the programs are in the WATFLOOD, Utilities and Flow Forecasting anuals
This space is left blank for your notes.

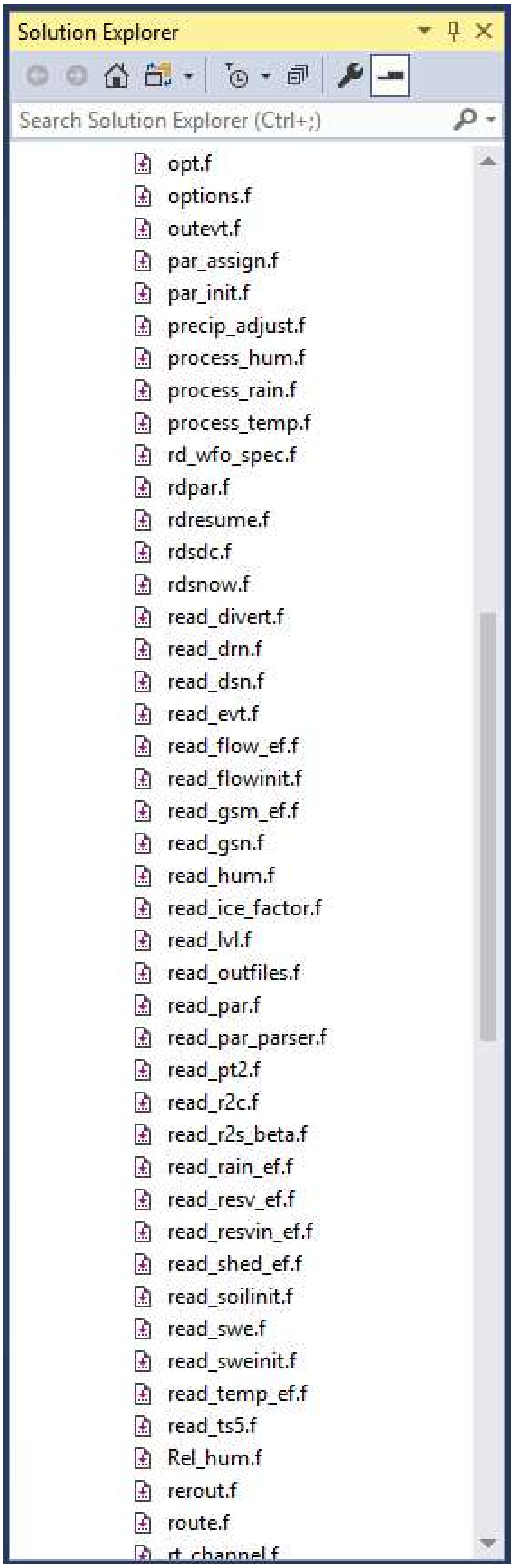
Note:
The WATFLOOD MS Visual Studio workspace is available on request. The Intel Fortran compiler is used.

The following CHARM components are not included in the release version of the MS Visual Studio workspace as these are either proprietary or copyrighted.. Calls to these subroutines are commented out in the various CHARM subroutines.

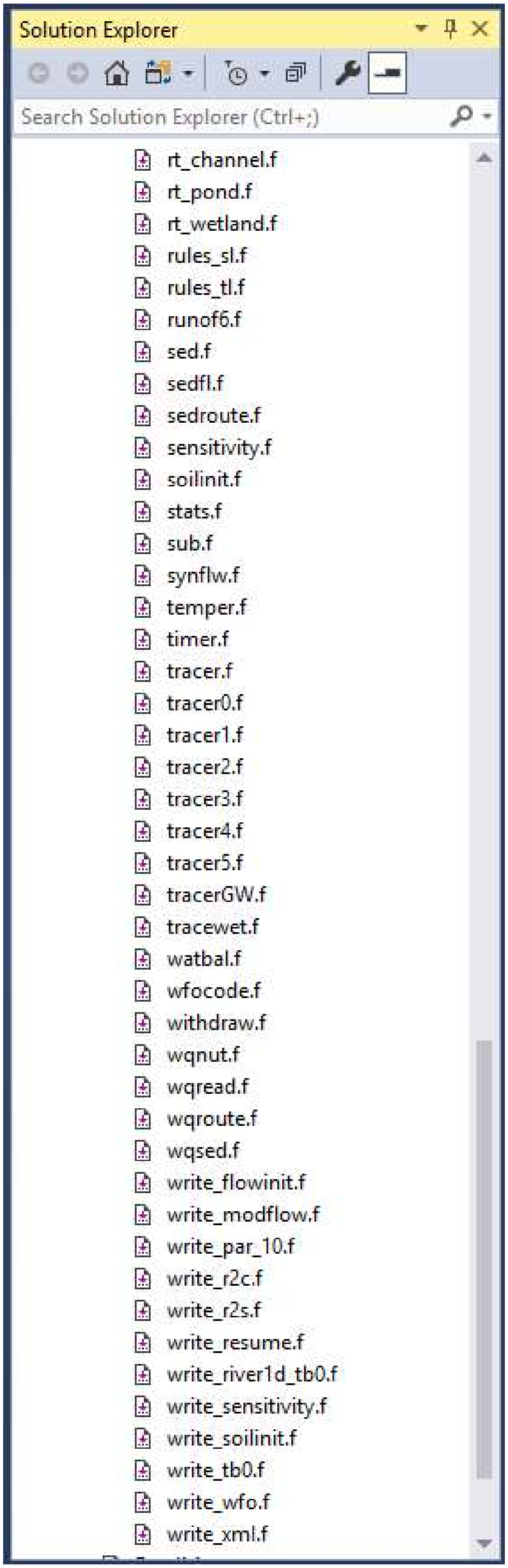
iso\ISO2Hlake.f
iso\ISO2Hland.f
iso\ISOlake.f
iso\ISOriver.f
iso\ISOwetland.f
iso\craig_gordon.f
iso\craig_gordon_2H.f
iso\isotopes.f
netCDF\read_2D_pcp_nc.F90
netCDF\read_2D_swe_nc.f90
netCDF\read_2D_tmp_nc.F90
netCDF\read_swe_date.f90
netCDF\read_swe_update.f90
netCDF\read_swe_use.f90
netCDF\read_ts_nc.F90
netCDF\read_uzs_update.f90
netCDF\write_2D_nc.f90
netCDF\write_ts_nc.f90



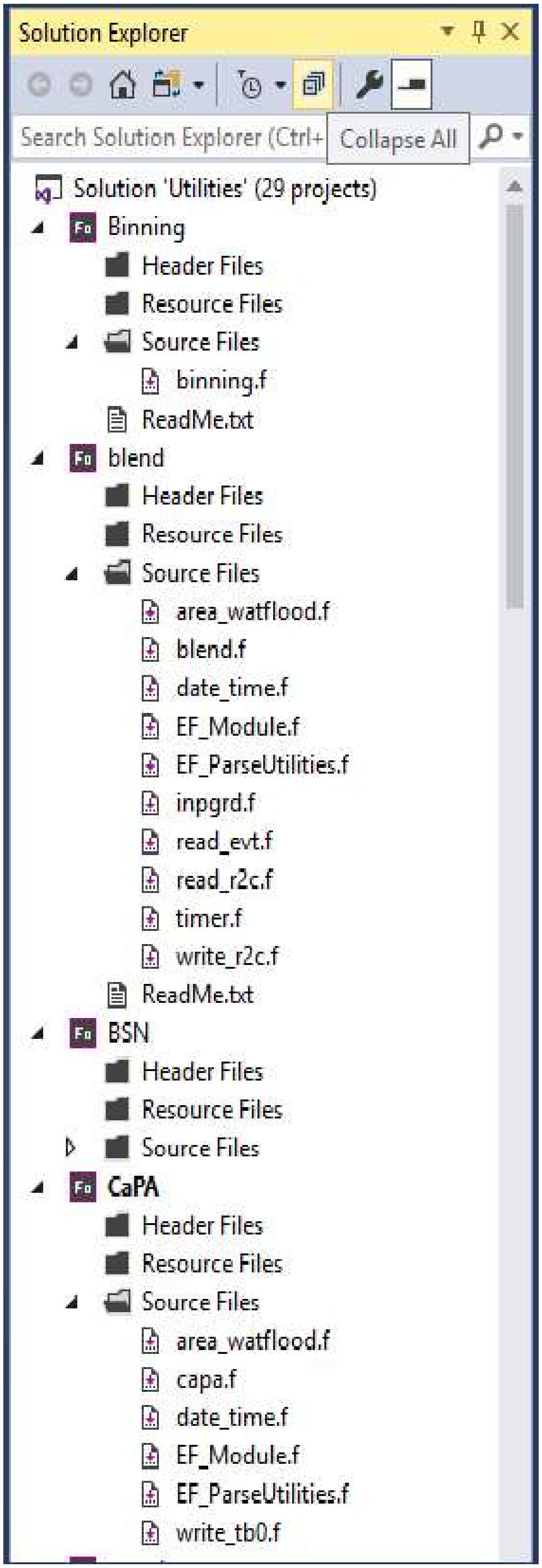
```
Solution 'CHARM' (2 projects)
  Charm
    Header Files
    Resource Files
    Source Files
      aet.f
      area_watflood.f
      Areacg.f
      baseflow.f
      CHARM.f
      craig_gordon.f
      craig_gordon_2H.f
      date_time.f
      dds_code.f
      EF_Module.f
      EF_ParseUtilities.f
      errormsg.f
      etharg.f
      etharg_beta.f
      etin.f
      etpriest.f
      filecode.f
      flowinit.f
      header.f
      ice_factor.f
      intcept.f
      ISO2Hlake.f
      ISO2Hland.f
      ISO2Hriver.f
      ISO2Hsnow.f
      ISO2Hwetland.f
      ISOlake.f
      ISOland.f
      ISOriver.f
      ISOsnow.f
      isotopes.f
      ISOwetland.f
      lake_evap.f
      lake_lvl_update.for
      lst.f
      melt.f
```



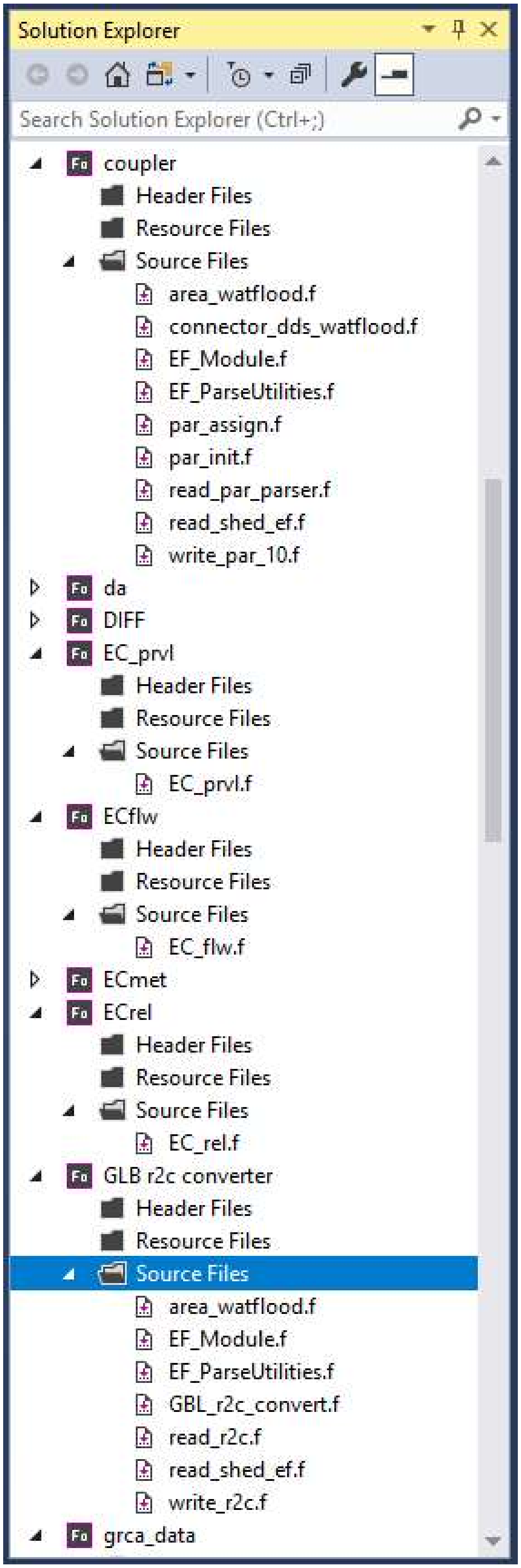
```
      opt.f
      options.f
      outevt.f
      par_assign.f
      par_init.f
      precip_adjust.f
      process_hum.f
      process_rain.f
      process_temp.f
      rd_wfo_spec.f
      rdpar.f
      rdresume.f
      rdsdc.f
      rdsnow.f
      read_divert.f
      read_drn.f
      read_dsn.f
      read_evt.f
      read_flow_ef.f
      read_flowinit.f
      read_gsm_ef.f
      read_gsn.f
      read_hum.f
      read_ice_factor.f
      read_lvl.f
      read_outfiles.f
      read_par.f
      read_par_parser.f
      read_pt2.f
      read_r2c.f
      read_r2s_beta.f
      read_rain_ef.f
      read_resv_ef.f
      read_resvin_ef.f
      read_shed_ef.f
      read_soilinit.f
      read_swe.f
      read_sweinit.f
      read_temp_ef.f
      read_ts5.f
      Rel_hum.f
      rerout.f
      route.f
      rt_channel.f
```



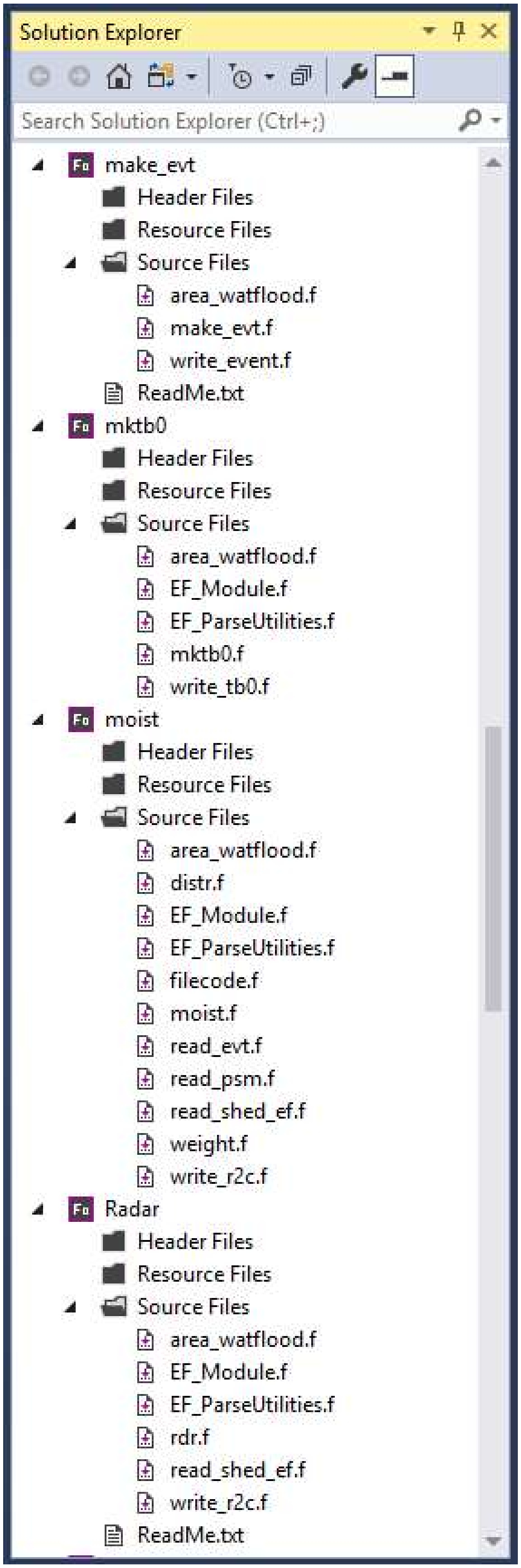
```
rt_channel.f
rt_pond.f
rt_wetland.f
rules_sl.f
rules_tl.f
runof6.f
sed.f
sedfl.f
sedroute.f
sensitivity.f
soilinit.f
stats.f
sub.f
synflw.f
temper.f
timer.f
tracer.f
tracer0.f
tracer1.f
tracer2.f
tracer3.f
tracer4.f
tracer5.f
tracerGW.f
tracewet.f
watbal.f
wfocode.f
withdraw.f
wqnut.f
wqread.f
wqroute.f
wqsed.f
write_flowinit.f
write_modflow.f
write_par_10.f
write_r2c.f
write_r2s.f
write_resume.f
write_river1d_tb0.f
write_sensitivity.f
write_soilinit.f
write_tb0.f
write_wfo.f
write_xml.f
```



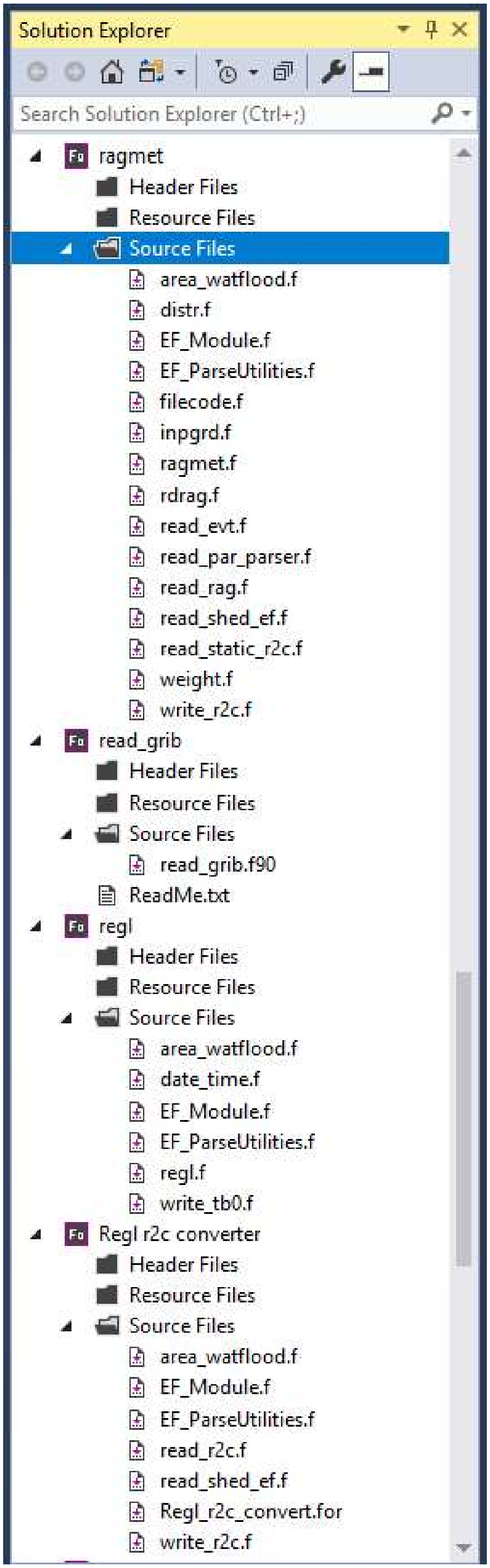
- Solution 'Utilities' (29 projects)
  - Binning
    - Header Files
    - Resource Files
    - Source Files
      - binning.f
    - ReadMe.txt
  - blend
    - Header Files
    - Resource Files
    - Source Files
      - area_watflood.f
      - blend.f
      - date_time.f
      - EF_Module.f
      - EF_ParseUtilities.f
      - inpgrd.f
      - read_evt.f
      - read_r2c.f
      - timer.f
      - write_r2c.f
    - ReadMe.txt
  - BSN
    - Header Files
    - Resource Files
    - Source Files
  - CaPA
    - Header Files
    - Resource Files
    - Source Files
      - area_watflood.f
      - capa.f
      - date_time.f
      - EF_Module.f
      - EF_ParseUtilities.f
      - write_tb0.f



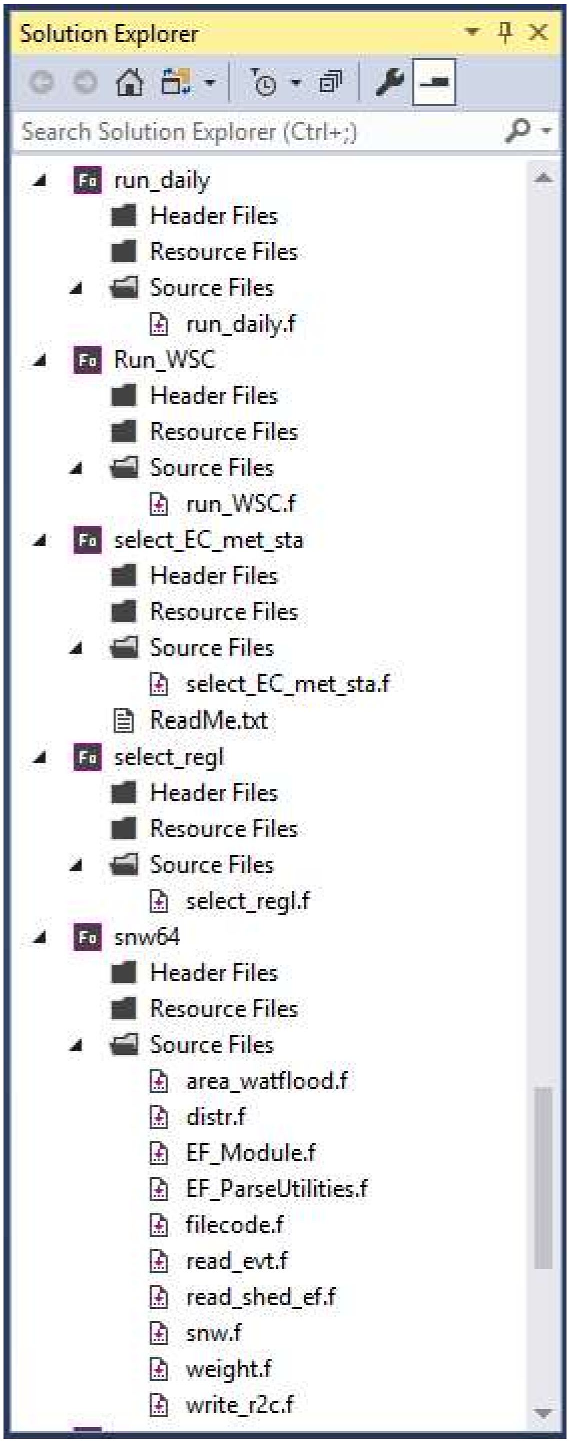
- coupler
  - Header Files
  - Resource Files
  - Source Files
    - area_watflood.f
    - connector_dds_watflood.f
    - EF_Module.f
    - EF_ParseUtilities.f
    - par_assign.f
    - par_init.f
    - read_par_parser.f
    - read_shed_ef.f
    - write_par_10.f
- da
- DIFF
- EC_prvl
  - Header Files
  - Resource Files
  - Source Files
    - EC_prvl.f
- ECflw
  - Header Files
  - Resource Files
  - Source Files
    - EC_flw.f
- ECmet
- ECrel
  - Header Files
  - Resource Files
  - Source Files
    - EC_rel.f
- GLB r2c converter
  - Header Files
  - Resource Files
  - Source Files
    - area_watflood.f
    - EF_Module.f
    - EF_ParseUtilities.f
    - GBL_r2c_convert.f
    - read_r2c.f
    - read_shed_ef.f
    - write_r2c.f
- grca_data



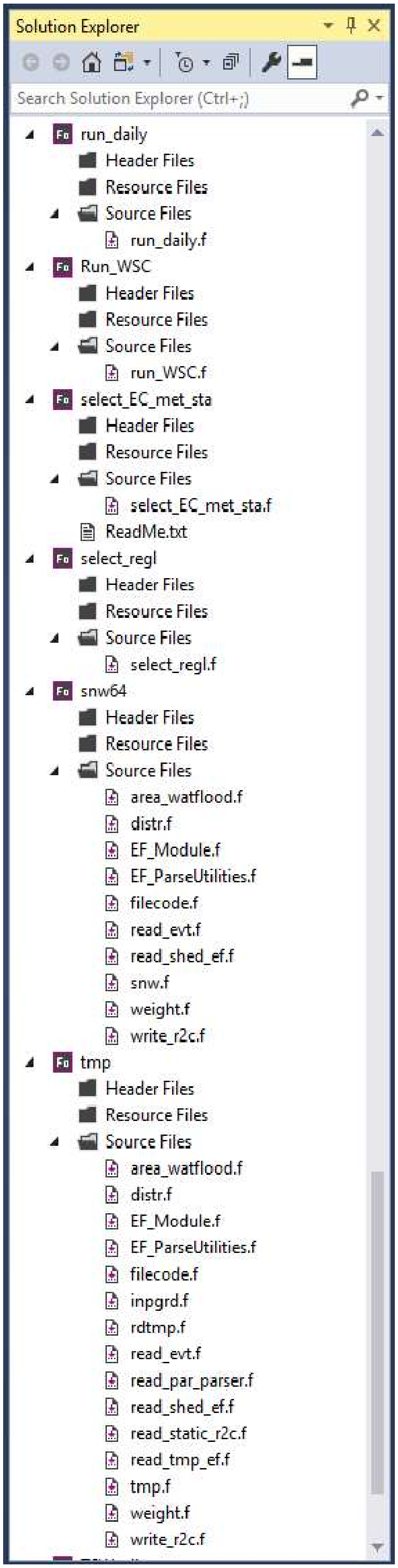
```
make_evt
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        make_evt.f
        write_event.f
    ReadMe.txt
mktb0
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        EF_Module.f
        EF_ParseUtilities.f
        mktb0.f
        write_tb0.f
moist
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        distr.f
        EF_Module.f
        EF_ParseUtilities.f
        filecode.f
        moist.f
        read_evt.f
        read_psm.f
        read_shed_ef.f
        weight.f
        write_r2c.f
Radar
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        EF_Module.f
        EF_ParseUtilities.f
        rdr.f
        read_shed_ef.f
        write_r2c.f
    ReadMe.txt
```



```
ragmet
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        distr.f
        EF_Module.f
        EF_ParseUtilities.f
        filecode.f
        inpgrd.f
        ragmet.f
        rdrag.f
        read_evt.f
        read_par_parser.f
        read_rag.f
        read_shed_ef.f
        read_static_r2c.f
        weight.f
        write_r2c.f
read_grib
    Header Files
    Resource Files
    Source Files
        read_grib.f90
    ReadMe.txt
regl
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        date_time.f
        EF_Module.f
        EF_ParseUtilities.f
        regl.f
        write_tb0.f
Regl r2c converter
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        EF_Module.f
        EF_ParseUtilities.f
        read_r2c.f
        read_shed_ef.f
        Regl_r2c_convert.for
        write_r2c.f
```

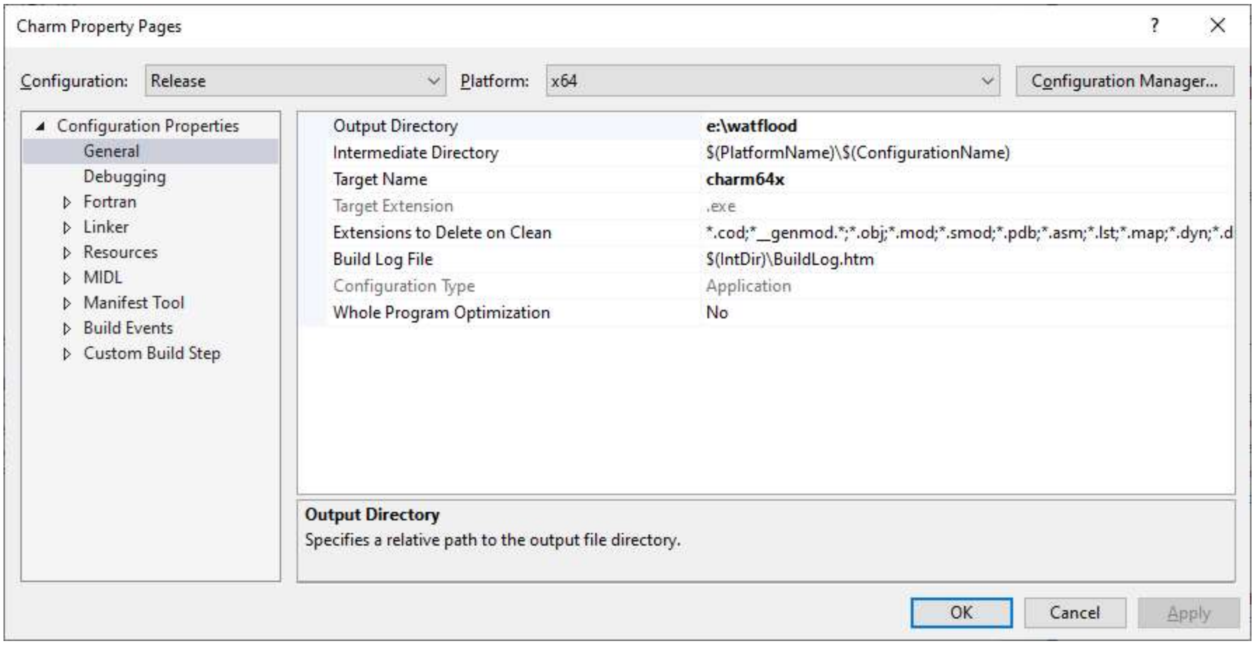


```
run_daily
    Header Files
    Resource Files
    Source Files
        run_daily.f
Run_WSC
    Header Files
    Resource Files
    Source Files
        run_WSC.f
select_EC_met_sta
    Header Files
    Resource Files
    Source Files
        select_EC_met_sta.f
    ReadMe.txt
select_regl
    Header Files
    Resource Files
    Source Files
        select_regl.f
snw64
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        distr.f
        EF_Module.f
        EF_ParseUtilities.f
        filecode.f
        read_evt.f
        read_shed_ef.f
        snw.f
        weight.f
        write_r2c.f
```



```
run_daily
    Header Files
    Resource Files
    Source Files
        run_daily.f
Run_WSC
    Header Files
    Resource Files
    Source Files
        run_WSC.f
select_EC_met_sta
    Header Files
    Resource Files
    Source Files
        select_EC_met_sta.f
    ReadMe.txt
select_regl
    Header Files
    Resource Files
    Source Files
        select_regl.f
snw64
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        distr.f
        EF_Module.f
        EF_ParseUtilities.f
        filecode.f
        read_evt.f
        read_shed_ef.f
        snw.f
        weight.f
        write_r2c.f
tmp
    Header Files
    Resource Files
    Source Files
        area_watflood.f
        distr.f
        EF_Module.f
        EF_ParseUtilities.f
        filecode.f
        inpgrd.f
        rdtmp.f
        read_evt.f
        read_par_parser.f
        read_shed_ef.f
        read_static_r2c.f
        read_tmp_ef.f
        tmp.f
        weight.f
        write_r2c.f
```

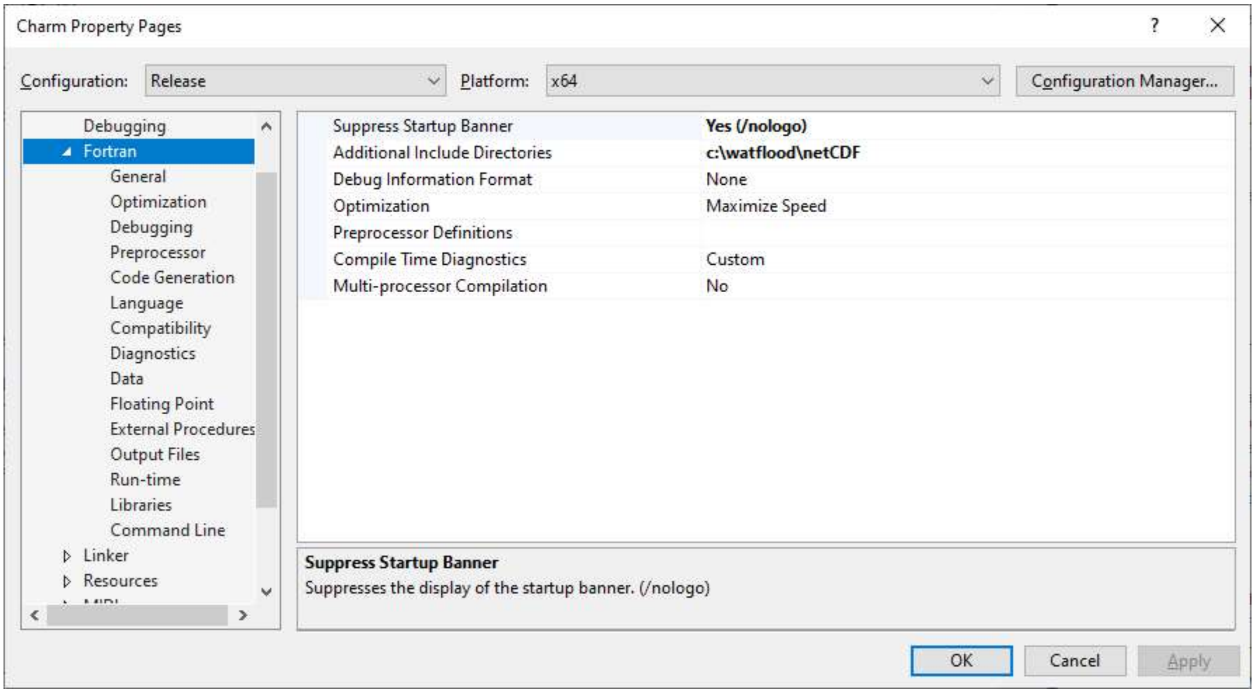
1.7 Visual Studio settings



Configuration: Release Platform: x64

Output Directory	**e:\watflood**
Intermediate Directory	$(PlatformName)\$(ConfigurationName)
Target Name	**charm64x**
Target Extension	.exe
Extensions to Delete on Clean	*.cod;*__genmod.*;*.obj;*.mod;*.smod;*.pdb;*.asm;*.lst;*.map;*.dyn;*.d
Build Log File	$(IntDir)\BuildLog.htm
Configuration Type	Application
Whole Program Optimization	No

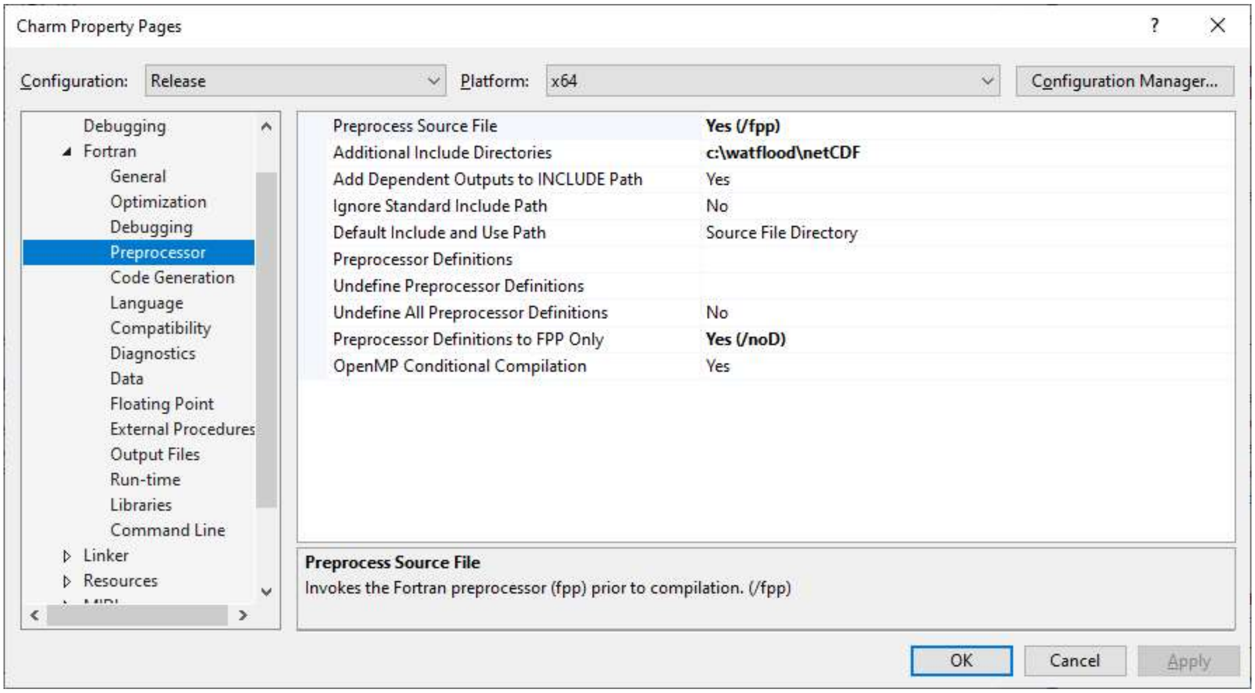
Output Directory
Specifies a relative path to the output file directory.



Configuration: Release Platform: x64

Suppress Startup Banner	**Yes (/nologo)**
Additional Include Directories	**c:\watflood\netCDF**
Debug Information Format	None
Optimization	Maximize Speed
Preprocessor Definitions	
Compile Time Diagnostics	Custom
Multi-processor Compilation	No

Suppress Startup Banner
Suppresses the display of the startup banner. (/nologo)

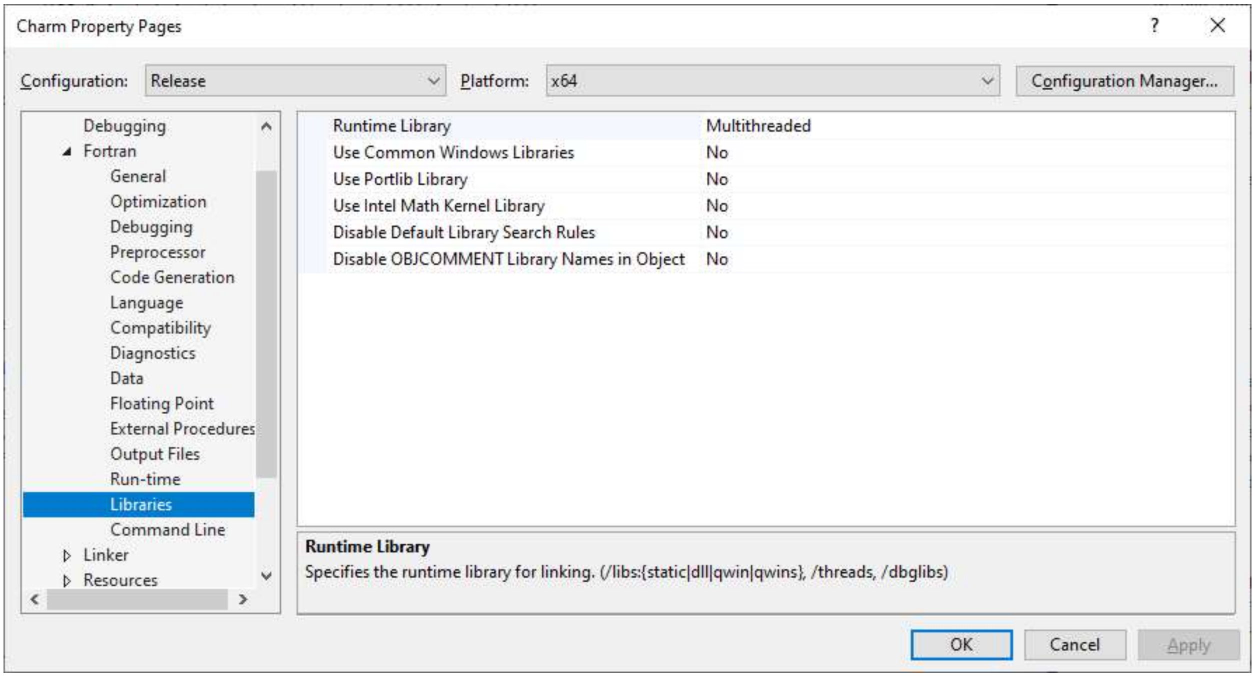


Charm Property Pages

Configuration: Release   Platform: x64

Preprocess Source File	**Yes (/fpp)**
Additional Include Directories	**c:\watflood\netCDF**
Add Dependent Outputs to INCLUDE Path	Yes
Ignore Standard Include Path	No
Default Include and Use Path	Source File Directory
Preprocessor Definitions	
Undefine Preprocessor Definitions	
Undefine All Preprocessor Definitions	No
Preprocessor Definitions to FPP Only	**Yes (/noD)**
OpenMP Conditional Compilation	Yes

Preprocess Source File
Invokes the Fortran preprocessor (fpp) prior to compilation. (/fpp)

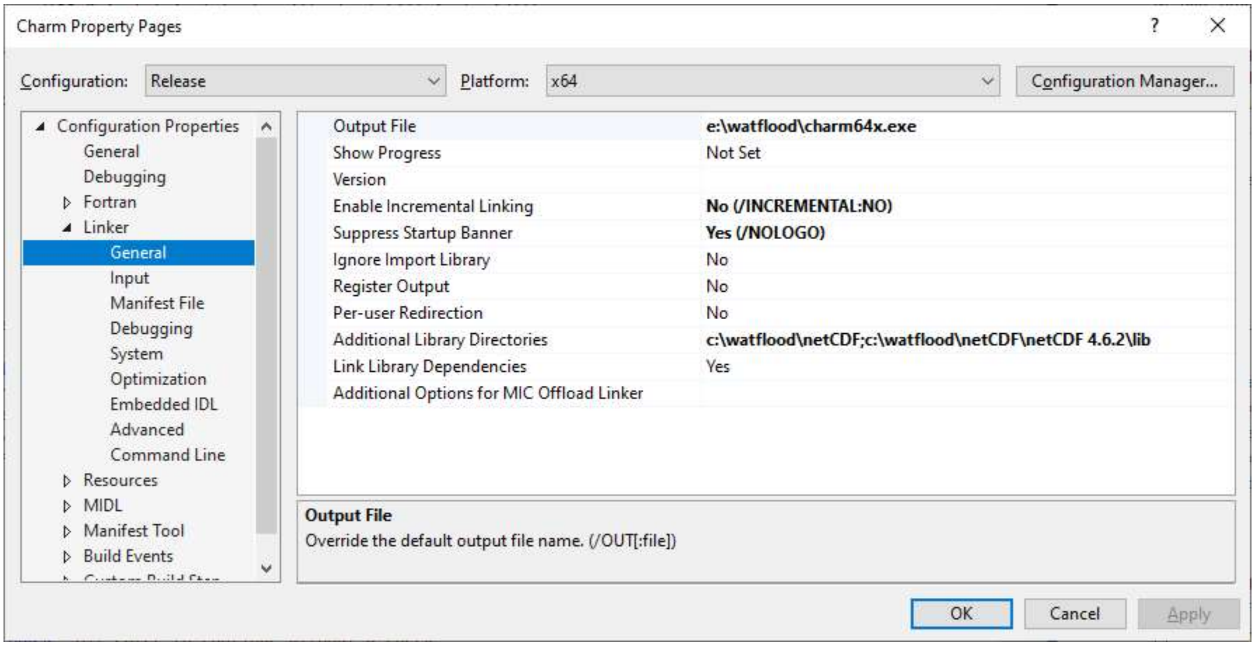


Charm Property Pages

Configuration: Release   Platform: x64

Runtime Library	Multithreaded
Use Common Windows Libraries	No
Use Portlib Library	No
Use Intel Math Kernel Library	No
Disable Default Library Search Rules	No
Disable OBJCOMMENT Library Names in Object	No

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

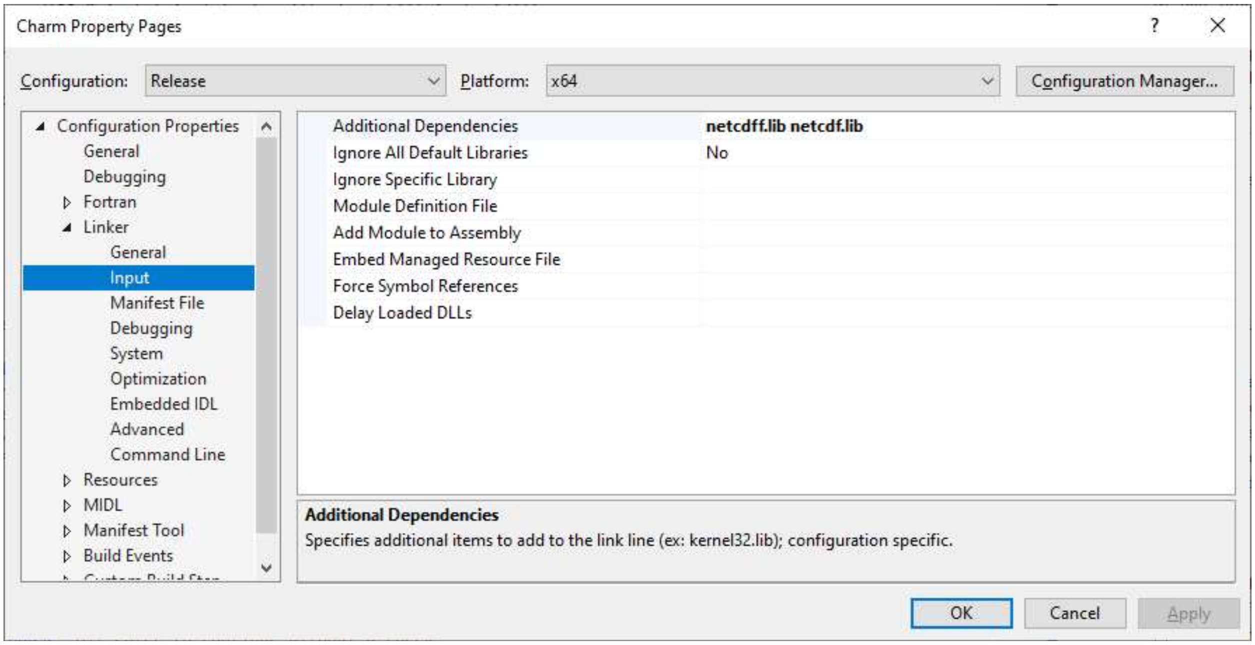


Charm Property Pages

Configuration: Release Platform: x64

Output File	**e:\watflood\charm64x.exe**
Show Progress	Not Set
Version	
Enable Incremental Linking	**No (/INCREMENTAL:NO)**
Suppress Startup Banner	**Yes (/NOLOGO)**
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**c:\watflood\netCDF;c:\watflood\netCDF\netCDF 4.6.2\lib**
Link Library Dependencies	Yes
Additional Options for MIC Offload Linker	

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



Charm Property Pages

Configuration: Release Platform: x64

Additional Dependencies	**netcdff.lib netcdf.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	

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

2 FEWS & netCDF additions to CHARM

2.1 Input files

New netCDF formats:

radcl\precip.nc	read_2D_pcp_nc.f90
tempr\tempt.nc	read_2D_tmp_nc.f90
strfw\flow.nc	read_ts_nc.f90
snow1\swe.nc	read_2D_swe_nc.f90

The usual required input files:

BSN.pdl
BSN_shd.r2c

Additional optional input files:

snow1\swe_update.xml	read_swe_update
snow1\swe_date.xml	read_swe_date
snow1\SWE_use.xml	read_swe_use
moist\uzs_update.xml	read_uzs_update

2.2 Output files

2.2.1 Time series 2-D output files – all code as separate s/r's in write_2D_nc.f90

results\CHARM_flow_2D.nc	write_2D_flow()
results\CHARM_grid_runoff_2D.nc	write_2D_grid_runoff()
results\CHARM_cumm_ET_2D.nc	write_2D_cumm_ET()
results\CHARM_swe_2D.nc	write_2D_swe()
results\CHARM_uzs_2D.nc	write_2D_uz()

Note: Each of the _write_2D_** files use "include write_2D_common,f90"

2.2.2 Time series 1-D output files – all code as separate s/r's in write_ts_nc.f90

results\CHARM_flow_vector.nc	write_ts_flow()
results\CHARM_lake_inflow.nc	write_ts_lake_inflow()
results\CHARM_lake_levels.nc	write_ts_lake_levels()
results\CHARM_lake_outflow.nc	write_ts_lake_outflow()

Note: Each of the _write_ts_** files use "include write_ts_common,f90"

3 DLL's for WATFLOOD

DLL's for CHARM

PROGRAM:	DLL:
Charm	hdf5.dll
Charm	hdf5_hl.dll
Charm	netcdf.dll
Charm	zlib1.dll
CHARM64d	libiomp5md.dll
OstrichMPI	msmpi.dll
OstrichMPI & Ostrich	msvcp120.dll
OstrichMPI & Ostrich	msvcr120.dll
Wget	libeay32.dll
Wget	libiconv2.dll
Wget	libintl3.dll
Wget	libssl32.dll
WGRIB2	cyggcc_s-seh-1.dll
WGRIB2	cyggfortran-3.dll
WGRIB2	cyggomp-1.dll
WGRIB2	cygquadmath-0.dll
WGRIB2	cygwin1.dll
	libmmd.dll
	ucrtbased.dll
	vcruntime140.dll
	vcruntime140d.dll

4 Arjen Markus correspondence (re: FEWS coupling using netCDF formats)

1/14/2019

Hello Nick,

Ivo asked me to look into the problem you are facing with incorporating NetCDF in the WATFLOOD adapter. Unfortunately the information I got is rather limited. I have ample experience with NetCDF in Fortran programs on both Linux and Windows, as we use it in our Delft3D software package. From comparing our solution for Delft3D with the list of source files you include I see that there are quite a few missing. Let me explain the set-up of the Fortran binding to NetCDF as I understand it:

- NetCDF has been implemented in C and for Windows/MSVC++ you can get the prebuilt libraries (as DLLs)
- The Fortran 90 binding as you use it is in fact layered on a binding for FORTRAN 77 and that part is missing in your solution
- So you need to add those files as well, nf_attio.F90, … see below for the complete list
- The one (import) library you need to add to the link step is netcdf.lib, but there is a umber of DLLs that your program will depend on – see below

I hope this will help.

Regards,

Arjen Markus

Source files to be included (copied from the vfproj file):

```
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\module_netcdf4_nc_interfaces.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\module_netcdf4_nf_interfaces.F90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\module_netcdf_nc_data.F90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\module_netcdf_nc_interfaces.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\module_netcdf_nf_data.F90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\netcdf4.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\netcdf4_func.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\netcdf4_visibility.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_attio.F90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_control.F90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_dim.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_genatt.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_geninq.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_genvar.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_misc.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_nc4.f90">
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_par_dummy.f90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_var1io.F90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_varaio.F90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_varmio.F90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_varsio.F90"/>
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\typeSizes.f90"/></Filter></Files>
```

The DLLs that you need with the program because of NetCDF:
netcdf.dll
hdf5.dll
hdf5_hl.dll
vcruntime140.dll

1/14/2019

Arjen:

Thank you very much for your help & nice to meet you.
The MS/VS workspace I sent had only the subroutines that I could see are needed but I have a more extensive list of files as attached. I am missing

```
<File RelativePath=".\netcdf-fortran-4.4.4\fortran\nf_par_dummy.f90"/>
```

But I will look for it on-line.
Ditto for the DLL's and lib
I will have more questions but just now I have one:
You say you have experience in both Linux & Windows. For Windows, do you use the MS Visual Studio and Intel Visual Fortran? I yes – I'm really reassured. If no, do you see my use of my VS + Intel VF as being a problem?
Regards, Nick
PS. I was born in Broek op Langedijk, NH. Moved to Canada as a kid in 1953.

I have attached the file you are missing – it is one we made ourselves because of the lacking functionality on Windows.

It took me a while to sort out which sources are actually needed, as we cannot really use the CMake build stuff within Delft3D. It would require users (Delft3D is open source) to work via CMake and the integration into the existing Delft3D VS solution is painful. And we also cannot supply a precompiled library as we do not know what compiler version the users will use. Anyway, the list I sent you is what works. I may have missed out on a number of DLLs you will find necessary by the way. The list is longer than what you actually use, because source files defining more than one routine lead to object files that have all these routines in them and they may have dependencies themselves.

Rest assured: on Windows we use the combination MS Visual Studio and Intel Fortran, so I am well acquainted with the peculiarities of that environment – slight sigh. (Linux has its own set of problems.)

Feb. 21, 2019

Hi Nick,

The error messages say that you are missing a file config.h. This is a file that is typically created by tools like autoconf and CMake to convey certain build parameters. If you copy the attached file to the directory netcdf-fortran-4.4.4\libsrc\, it should be found and that is likely to solve quite a few things. Hopefully even all. I modified that particular file manually.

As for the difference between NetCDF 4.6.1 and NetCDF4.6.2: that should be the C side of things only with no changes in the way routines are called. The libraries ought to be fine either way. So if you change the "1" into a "2", it should all work, though those are ominous words in the world of modern programming 😊.

Regards,

Arjen

From: Nicholas Kouwen <kouwen@uwaterloo.ca>
Sent: 18 January 2019 22:23
To: Arjen Markus <Arjen.Markus@deltares.nl>
Cc: Ivo Miltenburg <Ivo.Miltenburg@deltares.nl>; Edwin Welles <Edwin.Welles@deltares-usa.us>; Ben Balk <ben.balk@deltares-usa.us>
Subject: RE: Delft-FEWS WATFLOOD adapter - issue with NetCDF

Arjen:

Picking through the error list – I found the file you sent nf_par_dummy.f90 and it compiles ok

However, looking through the modifications for netCDF 4.6.2 , they seem pretty substantial and your make file is looking for netCDF 4.6.1

It could be that this is the problem.

I looked for netCDF 4.6.1 on Github but see only netCDF-C 4.6.1

Did a GitHub search: We couldn't find any repositories matching 'netCDF 4.6.1'

Nick
DISCLAIMER: This message is intended exclusively for the addressee(s) and may contain confidential and privileged information. If you are not the intended recipient please notify the sender immediately and destroy this message. Unauthorized use, disclosure or copying of this message is strictly prohibited. The foundation 'Stichting Deltares', which has its seat at Delft, The Netherlands, Commercial Registration Number 41146461, is not liable in any way whatsoever for consequences and/or damages resulting from the improper, incomplete and untimely dispatch, receipt and/or content of this e-mail.

1/12/2019

Hi Nick,

The error messages say that you are missing a file **config.h**. This is a file that is typically created by tools like autoconf and CMake to convey certain build parameters. If you copy the attached file to the directory netcdf-fortran-4.4.4\libsrc\, it should be found and that is likely to solve quite a few things. Hopefully even all. I modified that particular file manually.

As for the difference between NetCDF 4.6.1 and NetCDF4.6.2: that should be the C side of things only with no changes in the way routines are called. The libraries ought to be fine either way. So if you change the “1” into a “2”, it should all work, though those are ominous words in the world of modern programming 😊.

Regards,

Arjen

From: Nicholas Kouwen <kouwen@uwaterloo.ca>
Sent: 18 January 2019 22:23
To: Arjen Markus <Arjen.Markus@deltares.nl>
Cc: Ivo Miltenburg <Ivo.Miltenburg@deltares.nl>; Edwin Welles <Edwin.Welles@deltares-usa.us>; Ben Balk <ben.balk@deltares-usa.us>
Subject: RE: Delft-FEWS WATFLOOD adapter - issue with NetCDF

Arjen:

Picking through the error list – I found the file you sent **nf_par_dummy.f90** and it compiles ok

However, looking through the modifications for netCDF 4.6.2 , they seem pretty substantial and your make file is looking for netCDF 4.6.1

It could be that this is the problem.

I looked for netCDF 4.6.1 on Github but see only netCDF-C 4.6.1

Did a GitHub search: We couldn’t find any repositories matching 'netCDF 4.6.1'

Nick

21/1/2019

You're welcome – I was composing a review on an article ☺ and that was up to ten o'clock.

Regards,

Arjen

From: kouwen <kouwen@uwaterloo.ca>
Sent: 21 January 2019 21:55
To: Arjen Markus <Arjen.Markus@deltares.nl>
Subject: RE: Delft-FEWS WATFLOOD adapter - issue with NetCDF

Arjen:

Ok - perfect.
Thanks for helping out so late!

Nick

Sent from my Bell Samsung device over Canada's largest network.

-------- Original message --------
From: Arjen Markus <Arjen.Markus@deltares.nl>
Date: 2019-01-21 3:47 PM (GMT-05:00)
To: Nicholas Kouwen <kouwen@uwaterloo.ca>
Cc: Ivo Miltenburg <Ivo.Miltenburg@deltares.nl>, Edwin Welles <Edwin.Welles@deltares-usa.us>, Ben Balk <ben.balk@deltares-usa.us>
Subject: RE: Delft-FEWS WATFLOOD adapter - issue with NetCDF

Hi Nick,

No, with this library (netcdff.lib) you should be able to build the actual program via VS

Regards,

Arjen

From: Nicholas Kouwen <kouwen@uwaterloo.ca>
Sent: 21 January 2019 21:45
To: Arjen Markus <Arjen.Markus@deltares.nl>
Cc: Ivo Miltenburg <Ivo.Miltenburg@deltares.nl>; Edwin Welles <Edwin.Welles@deltares-usa.us>; Ben Balk <ben.balk@deltares-usa.us>
Subject: RE: Delft-FEWS WATFLOOD adapter - issue with NetCDF

Arjen:

So if I understand, I can now try to compile in MS Visual Studio / Intel Visual Fortran and just link in the lib we created?

Or does more library building need to be done?

Yes – there's progress!

Nick

From: Arjen Markus <Arjen.Markus@deltares.nl>
Sent: Monday, January 21, 2019 3:38 PM
To: Nicholas Kouwen <kouwen@uwaterloo.ca>
Cc: Ivo Miltenburg <Ivo.Miltenburg@deltares.nl>; Edwin Welles <Edwin.Welles@deltares-usa.us>; Ben Balk <ben.balk@deltares-usa.us>
Subject: RE: Delft-FEWS WATFLOOD adapter - issue with NetCDF

Hi Nick,

I would say: use this "patch" and try to build the library and then the sample program. We want to have a useable library 😊. (Somehow this parameter HAVE_TS29113_SUPPORT has wrongly become defined and therefore the fallback fails). Whatever the reason, we are making progress.

Regards,

Arjen