Guide for Installing CALMET/CALPUFF/CALWRF/CALPOST

by **Tim Chui** 23 Aug 2018
with initial contributions from **Matt Fung** 2016
and minor additions by **Roland Stull** 11 Sep 2018
for gfortran version 6.3.0

Dept. of Earth, Ocean & Atmospheric Sciences
Univ. of British Columbia, Vancouver, BC V6T 1Z4, Canada

For operating systems: Linux, Unix, MacOS

Prerequisites:

- Check that you have the GNU compilers installed on your system, specifically **gfortran**
  - **`which gfortran`**
    - Checks whether or not **gfortran** is in your PATH; if not, you need to add the compiler binary directory to your PATH:
      - **`export PATH=/path/to/compiler/bin:$PATH`**
      - Usually in `/usr/local/bin` or `/opt/local/bin`
    - If **gfortran** is not on your system, you'll need to install it; best way would be to use a package manager
  - **gfortran --version**
    - Checks compiler version
    - Should be version 5+ (i.e. 5.4.0, 6.1.0, 6.3.0, etc.)
- Check whether you have netCDF installed on your system; needed for CALWRF
  - **`which ncdump`**
    - Checks whether or not the netCDF utility **ncdump** is in your PATH; if not, you need to add the netCDF binary directory to your PATH:
      - **`export PATH=/path/to/netcdf/bin:$PATH`**
      - If you installed through a package manager, should be in `/opt/local/bin`
        Namely, the path is `/opt/local/bin/ncdump`
    - If netCDF is not installed on your system, you'll need to install it; best way would be to use a package manager, OR build netcdf-3.6.3 yourself
  - **`ncdump -V`** (may show version just by running **ncdump**)
    - Checks netCDF version
    - Either Version 3 or 4 work, but you have to keep know which version you're using (will apply later when installing CALWRF)
      (I have ncdump version 4.3.3.1 at the time of this writing.)

Source Files

- *CALMET_v5.8.5.zip*
  - Contains source code for CALMET v5.8.5 (USEPA approved)
- *CALMET.zip*
  - Contains source code for CALMET v5.8.4; also contains DEMO directory needed for tutorial run of CALMET v5.8.5

- *CALPUFF_v5.8.5.zip*
  - Contains source code for CALPUFF v5.8.5 (USEPA approved)
- *CALPUFF.zip*
  - Contains source code for CALPUFF v5.8.4; also contains DEMO directory needed for tutorial run of CALPUFF v5.8.5

- *CALWRF_v2.0.2_L131108.zip*
  - Contains source code for CALWRF v2.0.2
  - Requires netCDF for installation
  - ***User needs to provide own WRF meteorology output files in net CDF format [netCDF (wrfout)] files to test installation***

- *CALPOST_v6.221.zip*
  - Contains source code for CALPOST v6.221

CALMET

- Assume you're working in the home directory: create directory to hold all CAL-related files, and change directory to CAL
  - **`mkdir CAL`**
  - **`cd CAL`**

- Transfer or download all .zip files to CAL; can do so in Finder window (if on Mac), or by copying:
  - **`cp /path/to/CALPUFF.zip .`** (include the last space and period)
  - **`cp /path/to/CALMET.zip .`** (include the last space and period)
  - etc.

- Untar the CALMET zip files, either by double-clicking in the Finder window, or using a command-line utility (**tar, unzip, gzip,** depending on OS)
  - **`tar -xvzf CALMET.zip`**
  - **`tar -xvzf CALMET_v5.8.5.zip`**

- Change directory into source code directory of CALMET v5.8.5
  - **`cd CALMET_v5.8.5/CODE`**

- Open up *calutils.for* in an editor. I prefer BBEdit on the mac desktop. But if you prefer an editor in the terminal window, try
  - **`vi calutils.for`**

- In lines 2712 to lines 2715, add a **`!`** to the beginning of those lines to comment them out.

- Uncomment lines 2721 to 2724 by removing the **`c ***`** at the beginning of those lines (making sure that the first characters of the lines are several spaces from the leftmost column of the code)

- Refer to **`vi`** documentation (or whatever editor you're using) if you're unsure of how to make edits

- After making and saving your changes, the code should look like the following:

```
2706 c
2707 c ----------------------
2708 c --- Intel ifort and Lahey lf95 compilers:
2709 c     NARGS is an intrinsic function that returns the number of command-line
2710 c         arguments INCLUDING the command.
2711 c ----------------------
2712        !numargs=NARGS()
2713        !if(numargs.ge.1)then
2714        !   call GETARG(1,ctext)
2715        !endif
2716 c
2717 c ---------------------------------------------------------
2718 c --- Sun compiler, GNU gfortran, OR
2719 c --- HP compiler -- NOTE: HP needs +U77 switch on compile
2720 c ---------------------------------------------------------
2721        numargs=IARGC()
2722        if(numargs.ge.1)then
2723            call GETARG(1,ctext)
2724        endif
2725 c
2726 c
```

- These changes are needed because CALMET assumes users compile with Intel **ifort**, which contains the function NARGS() to determine how many command line arguments are used when calling **`calmet.exe`**; because we are using **gfortran**, the

function to use is instead IARGC()

- After you've made your changes, save and quit your editor (**:wq** if editing in **vi**).

- In your terminal window, type the following so that the **gfortran** compiler won't give a FATAL error due to a missing "m" flag:
  `export PATH=/usr/bin:$PATH`

- compile the code by creating the object **`calmet.exe`**
  - **`gfortran -o calmet.exe calmet.for`**
  - If you did it correctly, you should see a whole bunch of warnings (fine), but no errors (not fine)
- Assuming that the .exe was successfully built, run it:
  - **`./calmet.exe`**
  - Should see the following error:
    "At line 24501 of file calmet.for (unit = 15)
    Fortran runtime error: Cannot open file 'calmet.inp': No such file or directory"

- Copy over the DEMO file from *CALMET.zip* into the directory CALMET_v5.8.5 (at the same level as CODE):
  - **`cd ..`** (should now be in CALMET_v5.8.5)
  - **`cp –r ../CALMET/DEMO .`** (include the last space and period)
  - **`cd DEMO`**

- Link in the executable that you built in CODE into DEMO (you should now be in DEMO):
  - **`ln -s ../CODE/calmet.exe .`** (include the last space and period)

- *CMET.INP* is the input (control) file, and it specifies several .DAT files to run **`calmet.exe`;** the version you have in DEMO is already ready to go, and requires no editing

- You may see *cmet.dat* and *cmet.lst* in the directory; get rid of them to ensure that you produce your own versions after running CALMET:
  - **`rm cmet.dat`** (output from previous test run)
  - **`rm cmet.lst`** (log file from previous test run)

- Run **calmet.exe** with the input file:
  - **`./calmet.exe CMET.INP`**

- If all is well, you should see *cmet.dat* and *cmet.lst*, and **calmet.exe** was installed successfully. *cmet.lst* is viewable in a text editor, and I encourage you to examine it. *cmet.dat* is not viewable, but it is the input for CALPUFF.

CALPUFF

- Go back to the base installation directory
  - **`cd ~/CAL`**

- Untar the CALPUFF-related zip files by double-clicking on the file in the finder window. Or if you are in a terminal window, do the following.
  - **`tar -xvzf CALPUFF.zip`**
  - **`tar -xvzf CALPUFF_v5.8.5.zip`**
  - **`cd CALPUFF_v5.8.5/CODE`**

- In your favorite editor, Comment out lines 2723 to 2726, and uncomment 2732 to 2735, in *calutils.for*, identical to what you did for CALMET. Save your result. Once you're done, the code should look like the following:

```
2717 c
2718 c ----------------------
2719 c --- Intel ifort and Lahey lf95 compilers:
2720 c     NARGS is an intrinsic function that returns the number of command-line
2721 c         arguments INCLUDING the command.
2722 c ----------------------
2723        !numargs=NARGS()
2724        !if(numargs.ge.1)then
2725        !   call GETARG(1,ctext)
2726        !endif
2727 c
2728 c ------------------------------------------------------
2729 c --- Sun compiler, GNU gfortran, OR
2730 c --- HP compiler -- NOTE: HP needs +U77 switch on compile
2731 c ------------------------------------------------------
2732         numargs=IARGC()
2733         if(numargs.ge.1)then
2734             call GETARG(1,ctext)
2735         endif
2736 c
2737 c
```

- In the terminal window, type
  `export PATH=/usr/bin:$PATH`

- Compile **calpuff.exe**
  - **`gfortran -o calpuff.exe calpuff.for`**

- Once again, you should see warnings, but no errors. If you run calpuff:
  `./calpuff.exe`
  , you should see the following error:
  > "At line 40386 of file calpuff.for (unit = 1)
  > Fortran runtime error: Cannot open file 'calpuff.inp': No such file or directory"

- Like before, copy in the DEMO directory from *CALPUFF.zip* at the same level as CODE, change directory so that you're in DEMO, then link in your new executable:
  - **`cd ..`** (which should put you back in directory CALPUFF_v5.8.5)
  - **`cp -r ../CALPUFF/DEMO .`** (include the last space and period)
  - **`cd DEMO`**
  - **`ln -s ../CODE/calpuff.exe .`** (include the last space and period)

- The input file is *CPUF.INP*, the output log file is *CPUF.LST*, and the output files will be *CPUF.CON, CPUF.DRY, CPUF.VIS, CPUF.WET.* Get rid of any output files that might have already been provided, because you want to see if you can run CALUFF to create your own output files. So remove these files, either directly using the Mac Finder, or in your terminal window :
  - **`rm CPUF.LST`**
  - **`rm CPUF.CON`**
  - **`rm CPUF.DRY`**
  - **`rm CPUF.VIS`**
  - **`rm CPUF.WET`**

- The only .DAT file that CALPUFF needs to run is *cmet.dat*, which would have been produced from CALMET (the *cmet.dat* in DEMO was pre-made, and is not identical to the *cmet.dat* you produced when testing **calmet.exe**).

- Run CALPUFF:
  - **`./calpuff.exe CPUF.INP`**

- You should see the new log file *CPUF.LST*, and the new output files *CPUF.CON, CPUF.DRY, CPUF.VIS, CPUF.WET.* If all is well, **calpuff.exe** has been successfully installed. Only CPUF.LST is viewable in a text editor.

CALWRF

- The procedure to install CALWRF is very different from installing CALMET and CALWRF, and requires netCDF

- To view files above your home directory, type
  `cd ~/..`
  `ls`
  and use similar methods to navigate to places to see where your ncdump is.

- Set the environment variables to point to your netCDF libraries and include files
  - If your **ncdump** is in /path/to/netcdf/bin, then your libraries are likely in /path/to/netcdf/lib (should see *libnetcdff.a* and/or *libnetcdf.a*), and your include files are in /path/to/netcdf/include (should see *netcdf.inc*)
  - To answer the question above, one method is to use the Mac finder and search on **`ncdump`** to find which directory it is in.

- As an ***example***, if the root directory of your netCDF build is in /opt/local, then you should set the following variables:
  - **`export NETLIB=/opt/local/lib`**
  - **`export NETINC=/opt/local/inc`**
  - **`export NETINC=/opt/local/include`**

- Go back to the base installation directory
  - **`cd ~/CAL`**

- Unzip CALWRF in the Mac Finder by double clicking on it, or untar it in the terminal window:
  - **`tar -xvzf CALWRF_v2.0.2_L131108.zip`**

- In the terminal window, change your working directory
  - **`cd CALWRF_v2.0.2_L131108/code`**

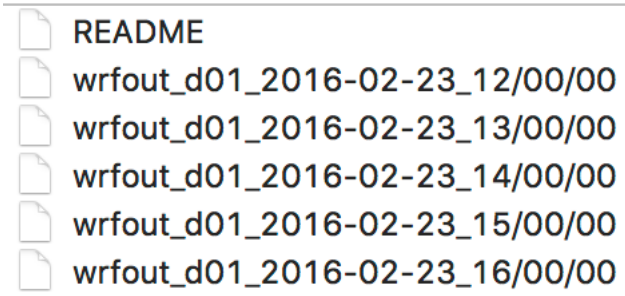
- Compile **calwrf.exe** depending on your version of netCDF (one line):
  - if you have netCDF-3:
    `gfortran -L$NETLIB -lnetcdf -I$NETINC -o calwrf.exe calwrf.f`
  - if you have netCDF-4:
    `gfortran -L$NETLIB -lnetcdff -I$NETINC -o calwrf.exe calwrf.f`

- The above commands mean to use gfortran, with netcdf libraries placed in $NETLIB (grabbing the appropriate file libnetcdf.a [netcdf-3; -lnetcdf] or libnetcdff.a [netcdf-4; -lnetcdf**f**]), as well as the header files included in $NETINC, and build the object file **calwrf.exe**

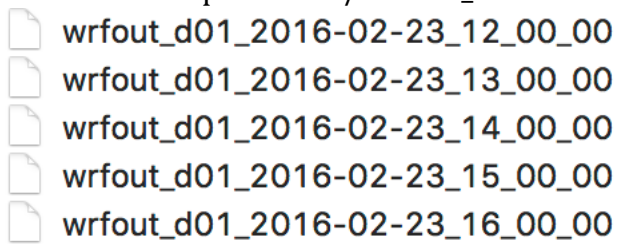
- If all is well, you should see no error, and the executable **calwrf.exe** should be produced.

- Next, you might need to edit the file names of the wrf output files (which you will use as input to CALWRF) to remove "/" from the file names, so the Mac doesn't accidentally think you are trying to go to a subdirectory.

- For example, for our course, we provided 5 output files from a WRF run, which are in a directory called `wrf_mini-outputFiles`



- Inside the **`CALWRF_v2.0.2_L131108`** directory, create a new directory called **`wrf_out`**, and copy the 5 wrf files into it. Then edit them using your favorite text editor to replace the "/" with "_" :



- The corresponding input file is *CALWRF_v2.0.2_L131108/calwrf.inp* , which was provided in the original files that you downloaded. It has a very nice set of comments showing how other file inputs might look. Please view their sample in a text editor.
  So you must edit their sample input file in a text editor to match whatever netCDF/wrfout file you wish to convert.

  - Here is how it looks, after I edited it to use the 5 wrf files shown above:

```
Create 3D.DAT file for WRF output
calwrf.lst          ! Log file name
calwrf.dat3         ! Output file name
-9,-9,-9,-9,-9,-9   ! Beg/End I/J/K ("-" for all)
-9                  ! Start datetime (UTC yyyymmddhh, "-" for all)
-9                  ! End   datetime (UTC yyyymmddhh, "-" for all)
5                   ! Number of WRF output files
wrf_out/wrfout_d01_2016-02-23_12_00_00 ! File name of wrf output (Loop over files
wrf_out/wrfout_d01_2016-02-23_13_00_00
wrf_out/wrfout_d01_2016-02-23_14_00_00
wrf_out/wrfout_d01_2016-02-23_15_00_00
wrf_out/wrfout_d01_2016-02-23_16_00_00

*****   Below are comments ***************************************
```

- Copy the **`calwrf.exe`** file that you produced into the main CALWRF... directory (i.e., the directory that already has the **`calwrf.inp`** file and the **`wrf_out`** subdirectory)

- Run calwrf:
  `./calwrf.exe`
  If successful, it will say "**`CALWRF succeeded`**" on your terminal window.

- The output of **calwrf.exe** will produce log file *calwrf.lst*, as well as a 2D file (10s of MB based on our 4 wrf files above) containing surface/diagnostic fields, and a 3D file (100s of MB) for use as input in CALMET.

CALPOST

- Go back to the base installation directory
  - **`cd ~/CAL`**

- Unzip CALPOST by double clicking on it in the Mac Finder. Or, in a termain window, Untar CALPOST:
  - **`tar -xvzf CALPOST_v6.221.zip`**
    and then change directories into it:
  - **`cd CALPOST/v6.221`**

- Using your favorite editor, Edit line 18078 in *CALPOST.FOR* so that the format string '**`(i)`**' has a specified width '**`(i121)`**', because
  - Unspecified widths are not supported in **gfortran**
  - Chose the number 121 for the width because the variable awork2 is 120 characters long at max (line 17997 is where awork2 is declared).

- Should look like the following after your edit:
-

```
18076 c ---    Extract receptor index
18077          call DEBLNK(awork1,1,ieq-1,awork2,n2)
18078          read(awork2(1:n2),'(i121)') irmap(nsamp)
18079 c ---    Extract sampler name
18080          call TLEFT(awork1,ieq+1,n1,awork2,n2)
```

- Comment out line 2597 in *calutils.for*; should look like the following after your edit:

```
2595
2596 c --- Get CPU time from SUN system utility (or PC dummy)
2597       !call etime(rcpu)
2598
2599       return
2600       end
```

- Comment out line 2713 in *calutils.for*, and uncomment lines 2735 to 2738; should look like the following after your edit:

```
2709 c
2710 c ------------------
2711 c --- Lahey compiler
2712 c ------------------
2713       !call getcl(ctext)
2714 c
2715 c ----------------------
2716 c --- COMPAQ DF compiler
2717 c ----------------------
2718 c ***  numargs=NARGS()
2719 c ***  if(numargs.ge.1)then
2720 c ***     call GETARG(1,ctext)
2721 c ***  endif
2722 c
2723 c ----------------------
2724 c --- Microsoft compiler
2725 c ----------------------
2726 c *** numargs=NARGS()
2727 c *** if(numargs.ge.1)then
2728 c ***    iarg=1
2729 c ***    call GETARG(iarg,ctext,istat)
2730 c *** endif
2731 c
2732 c ----------------
2733 c --- Sun compiler
2734 c ----------------
2735        numargs=IARGC()
2736        if(numargs.ge.1)then
2737           call GETARG(1,ctext)
2738        endif
2739 c
```

- Export the path: In the terminal window, type
  `export PATH=/usr/bin:$PATH`

- Compile **calpost.exe**
  - **`gfortran -o calpost.exe CALPOST.FOR`**

- You should see warnings, but no errors. If you run the executable
  `./calpost.exe`
  you should get the following error:
  "At line 708 of file CALPOST.FOR (unit = 4)
  Fortran runtime error: Cannot open file 'calpuff.con': No such file or directory"

- Go back to the previous directory, and run **calpost.exe** using the control file *Calpost.inp*, operating on the gridded file *CALPUFF.CON* (what you would produce if you ran CALPUFF)
  - **`cd ..`**
  - **`cp v6.221/calpost.exe .`**
  - **`./calpost.exe Calpost.inp`**

- CALPOST should produce a log file *calpost.lst*, which contains the gridded data from *CALPUFF.CON* in a readable format. You have now finished installing **calpost.exe.**

== end of guide ==