

Building the HWRF Components

Donald Stark

National Center for Atmospheric Research (NCAR)

The Developmental Testbed Center (DTC)

15 January, 2014

Outline

- HWRF system components
- Where to get the codes
- User directory structure
- Unpacking, setup, & build
- Build system infrastructure
- Support



HWRF System Components

- WRF: Weather Research & Forecast Model
- WPS: WRF Preprocessor
- UPP: Unified Post-processor
- GSI: Gridpoint Statistical Interpolation
- **HWRF-Utilities**: HWRF utilities & Vortex Init
- **POM-TC**: Tropical Cyclone version of the Princeton Ocean Model.
- **NCEP Coupler**: Couples WRF & POMTC
- **GFDL Vortex Tracker**: Storm tracker for post processing

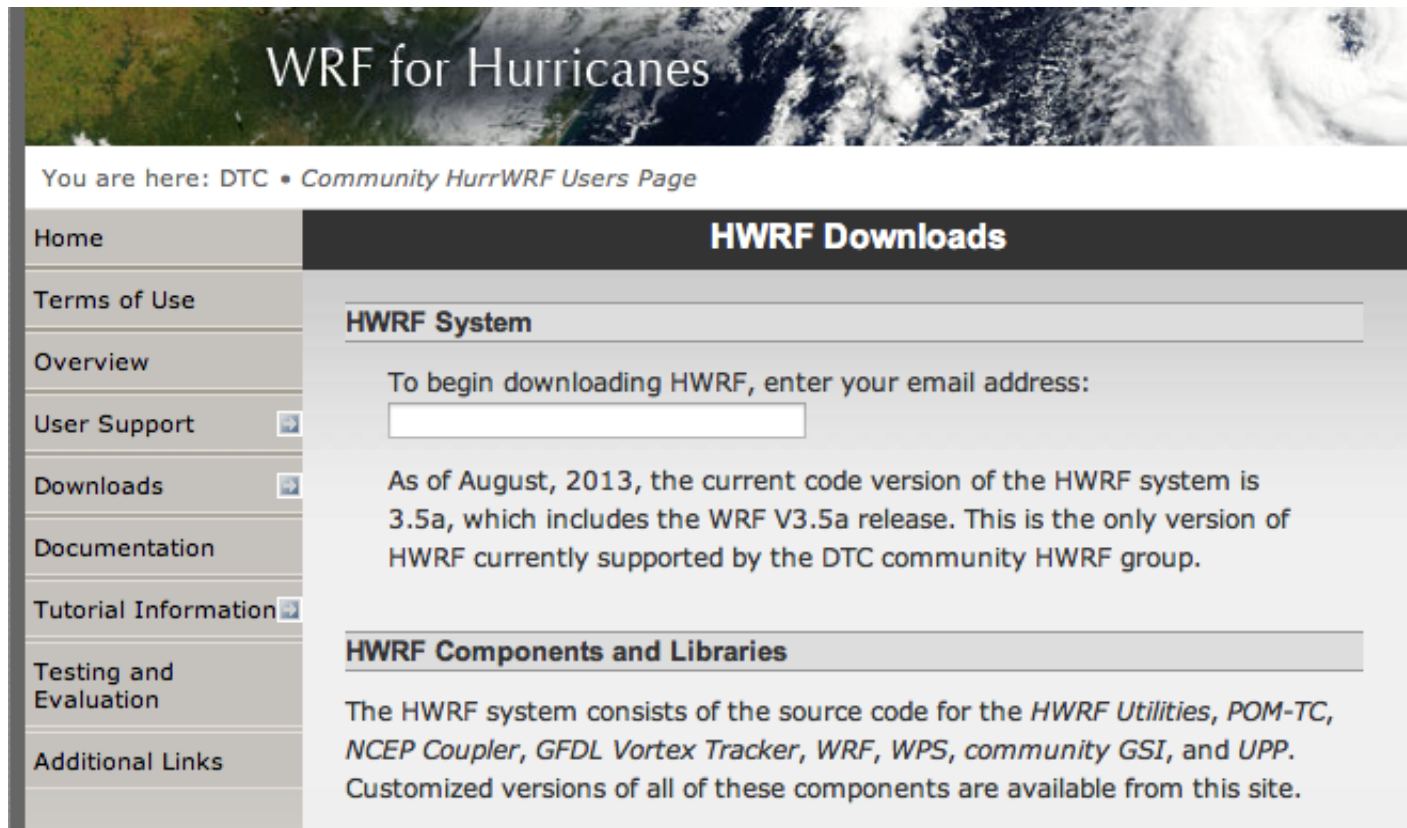


Downloading the code & setting up the directory structure



Downloading the Source Code

- All HWRF source code can be obtained from the WRF for Hurricanes web site:
 - www.dtcenter.org/HurrWRF/users/downloads/index.php



The screenshot shows the 'WRF for Hurricanes' website. The main heading is 'WRF for Hurricanes'. Below it, a breadcrumb trail reads 'You are here: DTC • Community HurrWRF Users Page'. A navigation menu on the left includes links for Home, Terms of Use, Overview, User Support, Downloads, Documentation, Tutorial Information, Testing and Evaluation, and Additional Links. The main content area is titled 'HWRf Downloads' and features a section for 'HWRf System' with a text input field for an email address. Below this, a paragraph states: 'As of August, 2013, the current code version of the HWRf system is 3.5a, which includes the WRF V3.5a release. This is the only version of HWRf currently supported by the DTC community HWRf group.' A second section, 'HWRf Components and Libraries', lists the following: 'The HWRf system consists of the source code for the *HWRf Utilities*, *POM-TC*, *NCEP Coupler*, *GFDL Vortex Tracker*, *WRF*, *WPS*, *community GSI*, and *UPP*. Customized versions of all of these components are available from this site.'



Setting up the HWRF Directory

1. Select the links to download the source code tar files:
 - [hwrfv3.5a_hwrf-utilities.tar.gz](#)
 - [hwrfv3.5a_pomtc.tar.gz](#)
 - [hwrfv3.5a_gfdl-vortextracker.tar.gz](#)
 - [hwrfv3.5a_ncep-coupler.tar.gz](#)
 - [hwrfv3.5a_WRFV3.tar.gz](#)
 - [hwrfv3.5a_WPSV3.tar.gz](#)
 - [hwrfv3.5a_UPP.tar.gz](#)
 - [hwrfv3.5a_GSI.tar.gz](#)
2. Create a working directory and move the src into it:
 - `mkdir ${HOME}/HWRF/src/`
 - `mv *.gz ${HOME}/HWRF/src/`
 - `cd ${HOME}/HWRF/src/`



Setting up the HWRF Directory

3. Move the tar files into the src directory created in 2.
4. Unpack the tar archives and clean up
 - *Uncompress the source code “gunzip *.tar.gz”*
 - *Untar the source code archives*
 - *Remove the tar files*
5. The src directory will contain the following subdirectories:
 - WRFV3
 - WPSV3
 - UPP
 - GSI
 - hwrf-utilities
 - gfdl-vortextracker
 - ncep-coupler
 - pomtc



Setting up the HWRF Directory

6. Check the “Release Notes” and “Known Issues” for code fixes needed prior to trying to build.

Check the release notes and known issues file for the most current information on a particular release.

Release notes: [\[Check\]](#)

Known issues: [\[Check\]](#)



Building the code



System Requirements

- C compiler
- Perl
- netCDF V3.6+ (for WRF, WPS, UPP & HWRF-Utilities)
- Linear algebra library (ESSL or LAPACK/BLAS)
- MPI V1.2+
- FORTRAN 90/95 compiler
 - IBM AIX (legacy support)
 - PGI pgf90
 - Intel ifort
 - Gfortran (limited support)
- GrADS (visualization)
- Grib 1 or 2 & wgrib (decode grib files)



Building the Components

- Code dependencies
 - WRF
 - WPS
 - UPP
 - GSI
 - HWRF-Utilities
 - HWRF Utilities
 - POM-TC
 - GFDL Vortex Tracker
- Order of build
 1. WRF
 2. WPS or UPP or GSI
 3. HWRF-Utilities
 4. Rest (POM-TC & Vortex Tracker & Coupler)



Building the Components

- Build sequence
 - ./clean -a
 - Set necessary environment variables & library paths
 - ./configure
 - Customize file *configure.component* if necessary
 - ./compile



Clean Compilation

- To remove all object files and executables, type:
clean
- To remove **all** built files, including the configure file, type: *clean -a*
 - A clean all needed if
 - Compilation failed
 - Want to change configuration file



Building WRF NMM

- Type *cd ./WRFV3*
- Type *./clean -a*
- Set library paths and system variables, for csh
 - *setenv HWRF 1*
 - *setenv NETCDF /path_to_netcdf_lib/*
 - *setenv WRF_NMM_CORE 1*
 - *setenv WRF_NMM_NEST 1*
 - *setenv WRFIO_NCD_LARGE_FILE_SUPPORT 1*
- For full details, see the WRF-NMM Users' Guide
 - <http://www.dtcenter.org/wrf-nmm/users/downloads/index.php>



Building WRF NMM (continued)

- Type *./configure*
 - Multiple options for Linux
 - For 64 bit PGI with gcc, select option 3 (**dmpar**)
 - For 64 bit PGI with pgcc (Yellowstone), select option 7 (**dmpar**)
 - For 64 bit PGI with pgcc (SGI MPT), select option 11 (**dmpar**)
 - For 64 bit Intel, select option 19 (**dmpar**)
 - etc.
 - Creates configuration file *configure.wrf*
- To compile type (for csh)
 - *./compile nmm_real |& tee wrf.log*
- Successful compilation will produce:
 - Two executables, *real_nmm.exe* & *wrf.exe*, in */main*



Building WRF Idealized Tropical Cyclone

- The Idealized Tropical Cyclone WRF creates two unique executables that can not be exchanged with the standard WRF executables. Therefore we create a separate directory for the Idealized WRF.
- After you expand the file `hwrfv3.5a_WRFV3.tar.gz`, change the directory name from `WRFV3` to `WRFV3_idealized`
- Type `mv ./WRFV3 ./WRFV3_idealized`
- Type `cd ./WRFV3_idealized`
- Set up the environment variables
 - `setenv HWRF 1`
 - `setenv NETCDF /path_to_netcdf_lib/`
 - `setenv WRF_NMM_CORE 1`
 - `setenv WRF_NMM_NEST 1`
 - `setenv IDEAL_NMM_TC 1`
 - `setenv WRFIO_NCD_LARGE_FILE_SUPPORT 1`



Building WRF Idealized Tropical Cyclone (continued)

- Type *./configure*
 - Select the appropriate build option for your platform
 - Creates configuration file *configure.wrf*
- To compile type (for csh)
 - *./compile nmm_tropical_cyclone |& tee wrf.log*
- Successful compilation will produce:
 - Two executables, *ideal.exe* & *wrf.exe*, in */main*
 - **NOTE:** *wrf.exe* created by setting the IDEAL_NMM_TC flag is **NOT** interchangeable with any other NMM WRF executables.



Building HWRF utilities

- Type `cd ./hwrf-utilities`
- Type `./clean -a`
- Set path to WRF (`WRF_DIR`) & check that netCDF library path `NETCDF` is set.
 - `setenv WRF_DIR ${HOME}/HWRF/src/WRFV3`
- Type `./configure`
 - Select compiler used to build WRF in previous step.
 - Creates file `configure.hwrf`
- For csh type `./compile |& tee hwrf-util.log`
- Successful compilation will produce:
 - 10 NCEPLib libraries in `hwrf-utilities/libs`
 - 24 executables in `hwrf-utilities/exec`



Building POMTC

- Type *cd ./pomtc*
- Type *./clean -a*
- Set library paths for **W3, SP, & SFCIO**.
 - *setenv LIB_SFCIO_PATH \${HOME}/HWRF/src/hwrf-utilities/libs/*
 - *setenv LIB_SP_PATH \${HOME}/HWRF/src/hwrf-utilities/libs/*
 - *setenv LIB_W3_PATH \${HOME}/HWRF/src/hwrf-utilities/libs/*
- And for LINUX, the **BLAS** library
 - *setenv LIB_BLAS_PATH \${HOME}/HWRF/src/hwrf-utilities/libs/*



Building POMTC (continued)

- Type *./configure*
 - Select compiler used to build WRF.
 - Creates configure file *configure.pom*
- For csh type *./compile |& tee pomtc.log*
- Successful compilation produces 13 executables
 - in *pomtc/ocean_exec*.



Building GFDL Vortex Tracker

- Type *cd ./gfdl-vortextracker*
- Type *./clean -a*
- Set library paths for **W3, SP, & BACIO**.
 - *setenv LIB_BACIO_PATH \${HOME}/HWRF/src/hwrf-utilities/libs/*
 - *setenv LIB_SP_PATH \${HOME}/HWRF/src/hwrf-utilities/libs/*
 - *setenv LIB_W3_PATH \${HOME}/HWRF/src/hwrf-utilities/libs/*



Building GFDL Vortex Tracker

- Type *./configure*
 - Select compiler used to build WRF.
 - Creates configuration file *configure.trk*
- Type *./compile |& tee tracker.log*
- Successful compilation produces three executables:
 - */trk_exec/hwrf_gettrk.exe*
 - */trk_exec/hwrf_tave.exe*
 - */trk_exec/hwrf_vint.exe*



Building NCEP Coupler

- Type *cd ./ncep-coupler*
- Type *./clean -a*
- Type *./configure*
 - Select compiler used to build WRF.
 - Creates configure file *configure.cpl*
- For csh type *./compile |& tee cpl.log*
- Successful compilation produces:
 - */cpl_exec/hwrf_wm3c.exe*



Building WPS

- Requires WRF, must be compiled first.
- Type *cd ./WPSV3*
- Type *./clean -a*
- Set the environment variable for the WRF path
 - *setenv WRF_DIR \${HOME}/HWRF/src/WRFV3*
- Type *./configure*
 - Choose the compile
 - Choose the serial option
 - Choose if you want GRIB2 support or not
 - Creates configuration file *configure.wps*
- For csh type *./compile |& tee wps.log*
- Successful compilation produces 3 executables:
 - in WPS/ *geogrid.exe*, *ungrib.exe*, & *metgrid.exe*
 - and a symbolic link to *mod_levs.exe* for use by the idealized tropical cyclone configuration



Building UPP

- Type *cd ./UPPV3*
- Type *./clean -a*
- Set library paths
 - *setenv WRF_DIR \${HOME}/HWRF/src/WRFV3*
 - *setenv HWRF 1*
- Type *./configure*
 - Select serial option
 - Creates configuration file *configure.upp*
- For csh type *./compile |& tee upp.log*
- Successful compilation produces 3 executables:
 - In bin/ *unipost.exe*, *ndate.exe*, & *copygb.exe*.



Building GSI

- Type *cd ./GSI*
- Type *./clean -a*
- Set library paths and system variables, for csh
 - *setenv HWRF 1*
 - *setenv WRF_DIR \${HOME}/HWRF/src/WRFV3*
- On some systems depending on the compiler setup (see Ch. 2 of the HWRF User's Guide for Details)
 - *setenv LAPACK_PATH /location of LAPACK library/*



Building GSI (continued)

- Type *./configure*
 - Select compiler used to build WRF in previous step.
 - Creates file *configure.gsi*
- For csh type *./compile |& tee gsi.log*
- Successful compilation will produce:
 - *GSI/run/gsi.exe*
 - *util/test/ssrc.exe*



Build Infrastructure



Build Infrastructure

- Uses DTC Build system based on WRF's.
- **/arch** directory contains rules & scripts for build.
 - **/arch/preamble**: uniform requirements for the code, such as word size, etc.
 - **/arch/configure.defaults** default platform settings
 - **/arch/postamble**: standard make rules & dependencies
- **./clean** script to clean the build.
- **./configure** script to create configuration file *configure.xxx*; contains info on compiler, MPI, & paths.
- **./compile** script to compile executable(s).



Summary & Support



Summary

- Download the tar files from the HWRF site.
- Don't mix components from different releases.
- Build components in proper order
 - i.e. WRF before HWRF-Utilities, WPS, UPP, & GSI.
 - i.e. HWRF-Utilities before POM-TC, coupler, & Vortex Tracker.
- Set necessary environment paths before running configure.
- When all else fails, read the Users' Guide.



Support

- For more detailed information on installation, please see:
 - HWRF Users' Guide
 - www.dtcenter.org/HurrWRF/users
 - WRF NMM Users' Guide
 - www.dtcenter.org/wrf-nmm/users/downloads/index.php
 - GSI User's Guide
 - www.dtcenter.org/com-GSI/users/index.php
- For further assistance contact:
 - wrfhelp@ucar.edu

