Compilation of WAVEWATCH III code

André van der Westhuysen

The WAVEWATCH III Team + friends
Marine Modeling and Analysis Branch
NOAA / NWS / NCEP / EMC

NCEP.list.WAVEWATCH@NOAA.gov
NCEP.list.waves@NOAA.gov
Covered in this lecture:

- How to install the model
- How does compiling of WAVEWATCH III work?
  - Why not direct FORTRAN code?
  - How does the pre-processor work?
Background material

- Background information can be found on the website below; particularly recommended are:
  - The WAVEWATCH III manual.
  - The best-practices guide for WAVEWATCH III programming.

  \[ \text{http://polar.ncep.noaa.gov/waves/wavewatch/wavewatch.shtml} \]

- Applications can be found on the NCEP operational wave model website below. Recommended information on this site are:
  - Two COMET training web sites.
  - Training material from this and previous courses.

  \[ \text{http://polar.ncep.noaa.gov/waves} \]
WAVEWATCH III Basics

WAVEWATCH III is a third-generation wave model descended from WAM, but with many differences:

- Governing equations (wave action vs. wave energy).
- Grid system (wavenumber on variable grid vs. relative frequency).
- Numerics (dynamic time stepping, higher-order propagation schemes, Garden Sprinkler Effect alleviation techniques).
- Physics (framework with multiple options).

WAVEWATCH I and II were developed at TU Delft (Netherlands) and NASA/Goddard Space Flight Center based on different governing equations.
WAVEWATCH III basics

WAVEWATCH III is similar to SWAN, but there are some major differences:

- Basic numerical solution techniques:
  - WAVEWATCH III: explicit hyperbolic equations, marching in time.
  - SWAN: implicit elliptical equations (stationary and nonstationary), solved by iterative sweeping.
  - However, both models now offer implicit propagation schemes on unstructured grids (coastal application)

- Program structure:
  - WAVEWATCH III: separate subprograms, compile-level user options
  - SWAN: single program, runtime user options

WAVEWATCH III v3.14 is trademarked and licensed, as close to open source as the US government allows.
WAVEWATCH III basics

WAVEWATCH III sub-programs

- Preprocessors:
  - Grid, numerics, physics: `ww3_grid`
  - Initial conditions, stationary BCs: `ww3_strt`
  - External boundary data: `ww3_bound`
  - Input fields: `ww3_prep`

- Core wave models:
  - `ww3_shel`, `ww3_multi`

- Output post-processors:
  - ASCII/Binary (point, field, track): `ww3_outp`, `ww3_outf`, `ww3_trck`
  - GRIB/GRIB2 (field): `ww3_grib`
  - NetCDF (point, field): `ww3_ounp`, `ww3_ounf`
  - GrADS (point, field): `gx_outf`, `gx_outp`
  - Wave system tracking (field): `ww3_systrk`

- Regridding processors: `ww3_gint`
Installing the code

Distribution methods

- Most WAVEWATCH III users will get the code from the NCEP website (after agreeing to the license).
  - Tar files + install script:
- Beta testers get newer versions.
- Those who work as co-developers have access to our Subversion (svn) server to get the latest developmental versions of the code.
  - svn server + install script.
- We will focus on the first distribution method; transitioning to second is trivial.

**NOTE:** WAVEWATCH III is Linux/UNIX only. Installation on windows will require preparations using Linux/UNIX.
## Installing the code

### WAVEWATCH III distribution files (old)

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>install_wwatch3</td>
<td>Installation script.</td>
</tr>
<tr>
<td>(install_ww3_svn)</td>
<td></td>
</tr>
<tr>
<td>wwatch3.aux.tar</td>
<td>Auxiliary programs and scripts, including GrADS scripts.</td>
</tr>
<tr>
<td>wwatch3.ftn.tar</td>
<td>Source code files.</td>
</tr>
<tr>
<td>wwatch3.inp.tar</td>
<td>Example input files, identical to those printed in the manual.</td>
</tr>
<tr>
<td>wwatch3.tst.tar</td>
<td>A large variety of test cases.</td>
</tr>
</tbody>
</table>
### Installing the code

#### WAVEWATCH III distribution files (current)

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>install_ww3_v4_svntar</td>
<td>Installation script.</td>
</tr>
<tr>
<td>(install_ww3_svn)</td>
<td></td>
</tr>
<tr>
<td>wwatch3.guide.tar</td>
<td>A guide for general coding practices</td>
</tr>
<tr>
<td>wwatch3.manual.tar</td>
<td>WAVEWATCH III manual</td>
</tr>
<tr>
<td>wwatch3.model.tar</td>
<td>The model with source codes, input files, auxiliary programs etc.</td>
</tr>
<tr>
<td>wwatch3.regtests.tar</td>
<td>A regression testing package that replaces the old set of tests</td>
</tr>
</tbody>
</table>
Installing the code

Installation procedure

- Copy five files to WAVEWATCH III designated directory, for instance ~/wwatch3
- Ensure that install_wwatch3 has execute permission.
- Execute the install script and answer all questions.
  - Will practice this in afternoon session.
  - Will require a basic FORTRAN 77 compiler to be assessable, typically gfortran (gnu) will do. This compiler will be used for aux programs only, not for actual WAVEWATCH III code.
- Add directories to search path in shell profile (e.g. .cshrc, .bashrc) as directed by script. In particular at paths for wwatch3/bin/ and wwatch3/exe/.
Installing the code

Installation procedure (2)

- NOTE: installation will generate file `.wwatch3.env` in the home directory.
- This file is used by all WAVEWATCH III management scripts, and points to directories used for code etc.

**HINT:** If multiple versions of WAVEWATCH III are maintained simultaneously, then:

- Point to proper directory by modifying `.wwatch3.env` by hand or by re-running `install_wwatch3`, or
- Place a generic name like `wwatch3/` in `.wwatch3.env` and use this as a symbolic link to the actual wave model directory (recommended).
Setting compiler options

- WAVEWATCH III is not distributed as ready-to-compile FORTRAN 90 code, but has a set of scripts to build the model according to user specifications:
- Critical files needed to compile:

<table>
<thead>
<tr>
<th>switch</th>
<th>List of model options selected by user (manual section 5.4). Preset with default model options.</th>
</tr>
</thead>
<tbody>
<tr>
<td>comp</td>
<td>Compile script (section 5.3). Requires user interventions once.</td>
</tr>
<tr>
<td>link</td>
<td>Link script (section 5.3). Requires user interventions once.</td>
</tr>
<tr>
<td>w3_make</td>
<td>Compiles wave model code-by-code.</td>
</tr>
</tbody>
</table>
Setting compiler options (2)

- The **compile** and **link** scripts need to be modified to address error capturing for the given hardware and compiler (see ‘comp=’ and ‘opt=’ lines).
- Setup procedure described in manual section 5.3.
- Various **comp** and **link** scripts are provided with the model distribution for well-known compilers (e.g. gfortran, ifort, pgf90).
- Please provide us with yours for further distribution with the code.

**NOTE:** the compiler used here can be different from the compiler set in `.wwatch3.env`, compile optimization is set in these scripts.
Installing the code

Setting compiler options (3)

- Installation on parallel systems (SMP or MPP) gets a little more complicated. Generally only the main programs `ww3_shel` or `ww3_multi` will be run as a parallel code, whereas all other codes remain serial. Proper compilation requires one of the following options:
  - First compile all auxiliary programs with the proper compile and link options (switches) of serial codes. Then reset compile and link options, and then compile `ww3_shel` and/or `ww3_multi` alone.
  - Generate individual and complete source codes for all programs and create the proper corresponding compile protocols.

See Day 4 presentation
Installing the code

Windows installation

- WAVEWATCH III is not set up for installation under MS Windows®.

- For installation under MS Windows, the following procedure can be used:
  - Find a Linux/UNIX box and perform the basic installation.
  - Set required model options in the `switch` file.
  - Run the script `w3_source` to extract the clean FORTRAN codes and corresponding `makefile` in tar files.
    - Set compiler options for MS Windows compiler in `w3_source`, or
    - manually edit makefiles as needed.
Installing the code

Subversion installation:

- Similar to installation from tar files, but …
  - Svn directory under main directory holds all versioned copies of model elements.
  - “Conventional” elements all are links to versioned copies.
  - Separate install script updates svn files, and all links.
  - Commit changes from svn directory, make sure new elements are added, old elements removed from here.
Coding philosophy

- WAVEWATCH III is intended as both:
  - A general modeling framework, and
  - An efficient operational wave model.

- This implies that…
  - It should be possible to include many options in the modeling framework, but
  - That the final compiled code should include only essential components.

To achieve this, the WAVEWATCH III source code is not plain FORTRAN 90, but needs to go through a preprocessor to obtain the FORTRAN 90 code.
### Filename convention

- WAVEWATCH III follows the FORTRAN 90 standard, and all files are build as complete modules to enable and enforce interface checking and use association.
- A typical WAVEWATCH III file is named `IDnamemd.ext`

<table>
<thead>
<tr>
<th>ID</th>
<th>Type identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>w3</td>
<td>Basic wave model routine</td>
<td></td>
</tr>
<tr>
<td>wm</td>
<td>Multi-grid extension routine</td>
<td></td>
</tr>
<tr>
<td>ww3_</td>
<td>Main program</td>
<td></td>
</tr>
<tr>
<td>gx_</td>
<td>GrADS postprocessor.</td>
<td></td>
</tr>
</tbody>
</table>

| name | Code name, typically 4 characters, e.g. “srce” |
| md   | Identifier that this is a module. |

<table>
<thead>
<tr>
<th>ext</th>
<th>File extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftn</td>
<td>Code requiring preprocessing</td>
<td></td>
</tr>
<tr>
<td>f90</td>
<td>Plain FORTRAN 90 code.</td>
<td></td>
</tr>
</tbody>
</table>
Filename convention (2)

- For instance,
  - `w3srcemd.ftn` contains the module of the basic wave model that processes source terms, and that will need to be preprocessed before it can be compiled.
  - `wmwavemdl.ftn` contains the multi-grid wave model module, requiring preprocessing.
  - `ww3_grid.ftn` contains the main program `ww3_grid` and requires preprocessing.
  - `mod_xnl4v5.f90` does not follow the convention, except that the file extension indicates that the file does not need to be preprocessed.
    - This is part of Gerbrant van Vledder’s quadruplet interaction package that is distributed with the wave model.
Using switches

- Below is part of `w3srcemd.ftn`, where input source terms are computed.
- The lines starting with `!/XXX` are optional pieces of code, activated by their “switches” `XXX`, in this case for
  - linear input, or
  - exponential input.

```fortran
! 2. Calculate source terms ----------------------------------------- *
! 2.a Input.

!/LN1        CALL W3SLN1 (       WN1, FHIGH, USTAR, U10DIR , VSLN       )
!/LNX        CALL W3SLNX
!/ST1        CALL W3SIN1 ( SPEC, WN2, USTAR, U10DIR ,        VSIN, VDIN )
!/ST2        CALL W3SIN2 ( SPEC, CG1, WN2, U10ABS, U10DIR, CD, Z0,        &
!/ST2                                                   FPI, VSIN, VDIN )
!/ST3        CALL W3SIN3 ( SPEC, CG1, WN2, U10ABS, USTAR, DAIR/DWAT, AS,  &
!/ST3                 U10DIR, Z0, CD, TAUWX, TAUWY, VSIN, VDIN, LLWS )
!/STX        CALL W3SINX
```
Using switches (2)

- Switches to be using in the compilation are stored in the `switch` file:
  - This file is stored as `/bin/switch` in the WAVEWATCH III directory. The installation script makes links to the original file in most work directories.
  - If the switches `NL1` and `ST2` are present in the `switch` file, the corresponding part of the preprocessed code `w3srcemd.f90` will become:

```fortran
! 2. Calculate source terms --------------------------------------------- *
! 2.a Input.
!
CALL W3SLN1 ( WN1, FHIGH, USTAR, U10DIR , VSLN )
!
CALL W3SIN2 ( SPEC, CG1, WN2, U10ABS, U10DIR, CD, Z0,         &
              FPI, VSIN, VDIN )
```

Manual Section 5.4
A FORTRAN 77 program `w3adc.f` is compiled during installation of the model to produce the program `w3adc`.

`w3adc` is managed by the script `ad3`, also put in place during model installation.

`ad3` also uses the `comp` script, in which compiler options are set.

`ad3` and the `link` script are called in the `makefile`, which is used by the standard UNIX/Linux `make` facility.

The `makefile` is updated by the script `make_makefile.sh`, every time the `switch` file is modified.

`make_makefile.sh` calls `w3_new` to touch the appropriate files to be recompiled by `make`.

And all this is managed by the `w3_make` script.
Compiling the code

Compilation flowchart

w3_make

make_makefile.sh

makefile

make (UNIX/Linux)

w3_new

ad3

link

w3adc

comp

Frequent interactive use

Possible interactive use

Manual chapter 5
Compiling the code

In summary

- Only `w3_make` is normally used:
  - `w3_make` by itself compiles all recognized WAVEWATCH III programs.
  - `w3_make ww3_grid` compiles this program only.
- `ad3` can be run interactively, particularly if test output needs to be switched on in selected routines.
- `make_makefile.sh` and `w3_new` can be run interactively as indicated in the manual.
- The rest of the system you will never see after the model is installed, but...

- It is essential that `comp` and `link` are set up with compiler error capturing if codes are to be edited.
Do’s and don’ts

- Even if the system may look a little complicated, do use it by properly modifying the `.ftn` files.
  - This is the only way of modifying this inside WAVEWATCH III in such a way that it can be ported to the distribution version of the model.
  - It is therefore more or less required by the license.

- `w3_source` will give you the clean FORTRAN files and the corresponding `makefile`:
  - Use this for operational implementations of the model.
  - Don’t use this for upgrading source code.
  - Use it for MS Windows applications …
Upon successful compilation, the following executables will reside in ./wwatch3/exe/ (slide 6):

- **Preprocessors:**
  - Grid, numerics, physics: `ww3_grid`
  - Initial conditions, stationary BCs: `ww3_strt`
  - External boundary data: `ww3_bound`
  - Input fields: `ww3_prep`

- **Core wave models:**
  - `ww3_shel, ww3_multi`

- **Output post-processors:**
  - ASCII/Binary (point, field, track): `ww3_outp, ww3_outf, ww3_trck`
  - GRIB/GRIB2 (field): `ww3_grib`
  - NetCDF (point, field) `ww3_ounp, ww3_ounf`
  - GrADS (point, field): `gx_outf, gx_outp`
  - Wave system tracking (field): `ww3_systrk`

- **Regridding processors:**
  - `ww3_gint`
The end

End of lecture