Logs Overview

HWRF Python Scripts Training

College Park, MD

January 22, 2016

Types of logs

- jlogfile
- Python standard error and standard output
- Per-job log files

A few common file locations

- **`$HOMEhwrf`** — HWRF installation directory
- **`$WORKhwrf`** — the directory in which each HWRF storm runs. There is one of these per cycle, per storm.
- **`$intercom`**=$WORKhwrf/intercom — a directory for trading data between jobs within one storm and cycle.
- **`$COMhwrf`** — the output directory for each cycle. There may be one of these per storm, or all storms may share one.

A few more common variables

- **`$log`** — log files that are not specific to a storm or cycle
- **`$job`** — the name of the job (post, forecast, products, etc.)
- **`$jobid`** — the job ID assigned by the batch system, or passed down to the scripts by ecFlow (NCO-specific)
- **`$YMD, $YMDH, $HH`** — components of the forecast cycle. For September 6, 2016, 00:00 UTC:
  - **`$YMD`** = 20160906
  - **`$YMDH`** = 2016090600
  - **`$HH`** = 00
- **`$STID`** — three-character storm id, such as 12L or 31W

NCO Variables

- **`$envir`** — NCO-specific variable: prod, para or test for the production, parallel or test version of HWRF.
- **`$stormnum`** — NCO-specific variable: a number from 1 to 7, for the storm priority.

Where are the logs?

- If you’re NCO:

```
$WORKhwrf=/tmpprd_p2/hwrf$stormnum_$envir_$HH/
$COMhwrf=/com2/hur/$envir/hwrf.$YMDH/
$log=/com2/output/$envir/$YMD/
$envir=prod
per-job logs: $log/hwrf$stormnum_$job.o$jobid
jlogfile: None?
```

- If you’re a repository user:

```
$WORKhwrf=$CDSCRUB/$SUBEXPT/$YMDH/$STID/
$COMhwrf=$CDSCRUB/$SUBEXPT/com/$YMDH/$STID/
$log=$CDSCRUB/$SUBEXPT/log/
per job logs: $WORKhwrf/hwrf_$job.log
jlogfile=$CDSCRUB/$SUBEXPT/log/jlogfile
```

jlog

- Located here for most: `pytmp/{EXPT}/log/jlogfile`
- NCO's jlog location is configured by `$jlogfile`
- Contains
  - A record of the completion of HWRF jobs
  - Log messages for all jobs run by that sub-experiment, for all storms and cycles.
- Only the highest-level messages are reported in the file
- To write to the jlogfile:
  - `produtil.log.jlogger.info`
  - `produtil.log.jlogger.critical`

jlogfile

```
11/05 21:12:22Z run_hwrf-INFO:  Successfully ran rocotorun for hwrf-Python_training-17W-2015082000.
11/05 21:20:05Z run_hwrf-INFO:  Successfully ran rocotorun for hwrf-Python_training-17W-2015082000.
11/05 21:21:20Z hwrf_launch_17W_2015082000_E99-INFO:  exhwrf_launch is starting
11/05 21:21:20Z hwrf_launch_17W_2015082000_E99-hwrf: ERROR:  /com/hur/prod/inpdata/nstorms: error reading: [Errno 2] No
such file or directory: '/com/hur/prod/inpdata/nstorms'.
 Will read all storms.
11/05 21:21:24Z hwrf_launch_17W_2015082000_E99-INFO:  ENS 99 (of 0) is not a perturbed ensemble member; not perturbing wind.
11/05 21:21:35Z hwrf_launch_17W_2015082000_E99-INFO:  exhwrf_launch completed
11/05 22:52:18Z run_hwrf-INFO:  Successfully ran rocotorun for hwrf-Python_training-17W-2015082000.
11/05 22:53:29Z hwrf_input_17W_2015082000_E99-INFO:  HWRF input job starting
11/05 22:53:39Z hwrf_input_17W_2015082000_E99-hwrf.exhwrf_input: ERROR:  [MainThread] Christina.Holt@dtn-
zeus.rdhpcs.noaa.gov: cannot access; will skip
11/06 00:09:32Z hwrf_input_17W_2015082000_E99-INFO:  HWRF input job completed
11/06 00:24:15Z run_hwrf-INFO:  Successfully ran rocotorun for hwrf-Python_training-17W-2015082000.
11/06 00:26:56Z hwrf_init_17W_2015082000_GFS_0_E99-INFO:  WPS Geogrid completed.
11/06 00:27:12Z run_hwrf-INFO:  Successfully ran rocotorun for hwrf-Python_training-17W-2015082000.
11/06 00:27:37Z hwrf_init_17W_2015082000_GDAS1_6_E99-INFO:  WPS Geogrid completed.
11/06 00:27:38Z hwrf_init_17W_2015082000_GDAS1_6_E99-INFO:  WPS Ungrib completed
11/06 00:27:43Z hwrf_init_17W_2015082000_GFS_0_E99-INFO:  WPS Ungrib completed
11/06 00:27:49Z hwrf_init_17W_2015082000_GDAS1_6_E99-INFO:  WPS Metgrid completed
11/06 00:28:02Z hwrf_init_17W_2015082000_GDAS1_9_E99-INFO:  WPS Geogrid completed.
11/06 00:28:03Z hwrf_init_17W_2015082000_GDAS1_3_E99-INFO:  WPS Geogrid completed.
11/06 00:28:05Z hwrf_init_17W_2015082000_GDAS1_3_E99-INFO:  WPS Ungrib completed
11/06 00:28:06Z hwrf_init_17W_2015082000_GDAS1_9_E99-INFO:  WPS Ungrib completed
11/06 00:28:15Z hwrf_init_17W_2015082000_GDAS1_3_E99-INFO:  WPS Metgrid completed
11/06 00:28:19Z hwrf_init_17W_2015082000_GDAS1_9_E99-INFO:  WPS Metgrid completed
11/06 00:28:27Z hwrf_init_17W_2015082000_GDAS1_6_E99-INFO:  fgat.t201508200000/realinit: completed
```

stderr and stdout

- Located in the $WORKhwrf directory
- stdout files contain all the logging (info, error, critical level) messages from the Python scripts
- stderr files contain all the error and critical messages, plus the submission information for the job (PROLOGUE, EPILOGUE)
- Separated into hwrf_*.out and hwrf_*.err. Name is set in Rocoto ent files.
- At least one set for each task.
- Multiple processor jobs have multiple sets of logs
  - post, products, tracker, erc.

Writing to the standard out

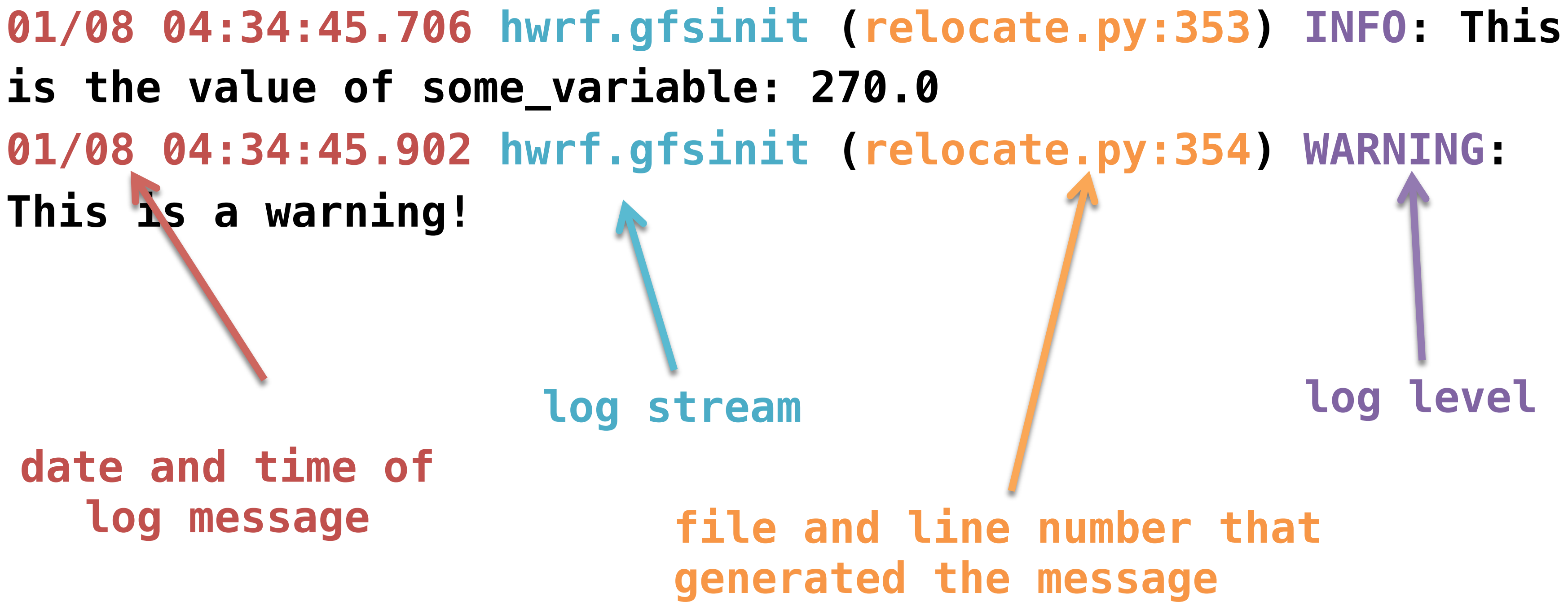
- Adding log messages can be done from the ush scripts with a few simple commands

```
logger=self.log()
logger.info(‘This is the value of some_variable:
              %s’ %(some_variable))
logger.warning(‘This is a warning!’)
logger.error(‘This is an error’)
logger.critical(‘This is really bad!’)
```

Result:

```
01/08 04:34:45.706 hwrf.gfsinit (relocate.py:353) INFO: This
is the value of some_variable: 270.0
01/08 04:34:45.902 hwrf.gfsinit (relocate.py:354) WARNING:
This is a warning!
```

Python log structure



```
01/08 04:34:45.706 hwrf.gfsinit (relocate.py:353) INFO: This
is the value of some_variable: 270.0
01/08 04:34:45.902 hwrf.gfsinit (relocate.py:354) WARNING:
This is a warning!
```

date and time of log message
log stream
file and line number that generated the message
log level

Python Logging Levels

	stdout	stderr	jlogfile	Meaning
DEBUG	N	N	N	Debug messages used by developer
INFO	Y	N	N	Regular status information
WARNING	Y	Y	N	Info useful for debugging failed jobs
ERROR	Y	Y	Y	Errors that degrade fcst or disable components
CRITICAL	Y	Y	Y	Failures that require intervention

Note: Log messages sent to the special "jlog" stream also go to the jlogfile, even if they're at lower log levels

Python Exception Stacks

- Several lines you get when HWRF components fail

```
Traceback (most recent call last):
  File "/pan2/projects/dtc-hurr/dtc/HWRF_training//scripts/
exhwrf_gsi.py", line 60, in <module>
    main()
  File "/pan2/projects/dtc-hurr/dtc/HWRF_training//scripts/
exhwrf_gsi.py", line 53, in main
    hwrf_expt.gsi_d02.run()
  File "/pan2/projects/dtc-hurr/dtc/HWRF_training/ush/hwrf/gsi.py", line
982, in run
    self.grab_enkf_input()
  File "/pan2/projects/dtc-hurr/dtc/HWRF_training/ush/hwrf/gsi.py", line
285, in grab_enkf_input
    self.grab_gfs_enkf()
  File "/pan2/projects/dtc-hurr/dtc/HWRF_training/ush/hwrf/gsi.py", line
607, in grab_gfs_enkf
    %(there,))
GSIInputError: required input file is empty or non-existent: /pan2/
projects/dtc-hurr/dtc/HWRF_training/pytmp/HWRF_training/2015082000/17W/
hwrfdata/enkf.2015081918/sfg_2015081918_fhr06s_mem001
```

Logs from components

- Many components have their own special log files
- For example:
  - WPS: metgrid.log.*, geogrid.log.*, ungrib.log
  - GSI: stdout
  - Coupler: cpl.out
  - WRF: rsl.out.* and rsl.err.*

Forecast Logs

- Three coupled components: Atmosphere, Ocean, Coupler
- Coupler and ocean share `$WORKhwrf/cpl.out`
  - This extra file exists because it's huge
- WRF has an out and err file for each rank
  - `$WORKhwrf/runwrf/rsl.out.RANK`
  - `$WORKhwrf/runwrf/rsl.err.RANK`
- The WRF master process does extensive logging in `$WORKhwrf/runwrf/rsl.out.0000`
  - Note: A failure could occur in any rank, and would be in that rsl.err or rsl.out file

Post-processing & Regribbing Logs

- Post-processing is split into post and products jobs
  - post runs the UPP to convert WRF output files to native e-grid GRIB files
  - products regrids the UPP output to standard grids, copies the GRIB files and native WRF output files to `$COMhwrf`, and runs the GFDL vortex tracker
- The post standard out is very large and is deleted upon success of post. If there is a failure, the log lives here:

```
$WORKhwrf/post.* /vpost.log
```

Products Logs

- Gribbers — these perform regribbing operations on post output.
  - Runs cnvgrib, wgrib and hwrf_egrid2latlon (copygb) programs.

```
$WORKhwrf/$jobid-gribber[1-7].log
```

- Copiers — these copy native model output. Once all native model output is copied, they start regribbing instead.

```
$WORKhwrf/$jobid-copier.log
```

- Trackers — these run the GFDL Vortex Tracker on outputs from the Gribbers.

```
Main Tracker: $WORKhwrf/$jobid-tracker.log
d02 Tracker: $WORKhwrf/$jobid-d02tracker.log
d01 Tracker: $WORKhwrf/$jobid-d01tracker.log
```

Init & Bdy Logs

- These jobs run many programs with extensive logging. Each init job has its own Python standard output/error stream, but each program also generates logs
- Two types of initialization: gfsinit and fgatinit

`$WORKhwrf/gfsinit` — parent global model full-length forecast. By default, this is the GFS.

`$intercom/gfsinit` — intercom delivery location for that init

`$WORKhwrf/fgat.$YMDH00` — parent global model short-length forecast for analysis. By default, this is the GDAS. There are usually three of these, for the HWRF analysis time -3, +0 and +3 hours.

`$intercom/fgat.$YMDH00` — intercom delivery location for that init

Init & Bdy Logs

- Geogrid
  - stdout/stderr — `wps/geogrid.log`
  - per rank — `wps/geogrid.log.RANK`
- Ungrib
  - stdout/stderr — `wps/ungrib.log`
- Metgrid
  - stdout/stderr — `wps/metgrid.log`
  - per rank — `wps/metgrid.log.RANK`
- prep_hybrid
  - While running: `$WORKhwrf/(init)/prep_hybrid/$YMDH/prep.log`
  - When finished: `$intercom/(init)/prep_hybrid/prep_$piece.log` where $piece is the boundary time index.

Init & Bdy Logs

- WRF and Real: (see Forecast logs)
  - init-length real_nmm — realinit/
  - forecast-length real_nmm — realfcst/
  - wrfanl run of the wrf — wrfanl/
  - ghost run of the wrf — ghost/
- For generating parent vortex location: (see Post-processing Logs)
  - post (while running or if failed) — post.*
  - hwrf.gribtask to convert to lat-lon — regribber/
  - tracker — tracker/

Questions?

Up next…

Troubleshooting