

---

# Quickstart Guide: FMS AM2 Model

Amy Langenhorst <Amy.Langenhorst@noaa.gov>

The AM2 package contains source code and scripts for GFDL's atmosphere-only experiment using observed sea surface temperatures, time varying radiative forcings (including volcanos) and time varying land cover type. This version of AM2 (also called AM2.1) uses the finite-volume dynamical core (Lin 2004). The test case included with this release uses the same cloud tuning, radiative forcing and land cover type as the coupled model (Delworth et al. 2006). Several bug fixes are included that increase the stability of the model but do not appreciably change the climate from the previous version (GAMDT 2004). The resolution is 2.5 degrees longitude by approximately 2 degrees latitude (M45) with 24 vertical levels (L24). The model version is AM2p14. This document describes the available documentation and reference material for the FMS AM2 model.

## References:

The GFDL Global Atmospheric Model Development Team, 2004: **The new GFDL global atmosphere and land model AM2-LM2: Evaluation with prescribed SST simulations.**

*Journal of Climate*, **17(24)**, 4641-4673. **Abstract** [<http://www.gfdl.noaa.gov/reference/bibliography/2004/gamdt0401.html>] / **PDF** [<http://www.gfdl.noaa.gov/reference/bibliography/2004/gamdt0401.pdf>]

Delworth, T. L., et al., 2006: **GFDL's CM2 Global Coupled Climate Models. Part I: Formulation and simulation characteristics.** *Journal of Climate*, **19(5)**, 643-674.

Lin, S-J., 2004: A "vertically Lagrangian" finite-volume dynamical core for global models. *Monthly Weather Review*, **132(10)**, 2293-2307.

## Table of Contents

1. Acquire the Source Code .....	1
2. Acquire the Input Datasets .....	2
3. Run the Model .....	2
3.1. Functionality of the Sample Runscripts .....	2
3.2. Portability Issues with the Sample Runscripts .....	3
3.3. Changing the Sample Runscripts .....	3
4. Examine the Output .....	3

## 1. Acquire the Source Code

The Flexible Modeling System development team at GFDL uses a local implementation of GForge to serve FMS software, located at <http://fms.gfdl.noaa.gov>. In order to obtain the source code, you must **register** [<https://fms.gfdl.noaa.gov/account/register.php>] as an FMS user on our software server. After submitting the registration form on the software server, you should receive an automatically generated confirmation email within a few minutes. Clicking on the link in the email confirms the creation of your account.

After your account has been created, you should **log in** [<https://fms.gfdl.noaa.gov/account/login.php>] and request access to the FMS AM2 Model project. Once the FMS project administrator grants you access, you will receive a second e-mail notification. This email requires action on the part of the project administrator and thus may take

longer to arrive. The email will contain a software access password along with instructions for obtaining the release package, which are described below.

You may download the tar file called `memphis_src.tar.gz` from <https://fms.gfdl.noaa.gov/projects/am2/>. Input datasets and sample output is also available there for download. See [Section 4, “Examine the Output”](#) for more information on the sample output.

## 2. Acquire the Input Datasets

You will need to download the following files to use as input data for the AM2 model.

**Table 1. Input Datasets**

File	Size	Content
<code>memphis_input_data.tar.gz</code>	2.5 GB	Input datasets for physical parameterizations
<code>M45.tripolar.grid_spec.nc.gz</code>	3.2 MB	Grid spec file (exchange grid) for use with Finite-Volume M45 model
<code>M45L24_Haloe.tar.gz</code>	15 MB	Finite-Volume M45 model initial conditions

You can download these datasets from GFDL's Gforge server at <https://fms.gfdl.noaa.gov/projects/am2/>.

Once you have downloaded the files and placed them into the directory in which you want to store them, edit the variables `inputData`, `gridSpec` and `initCond` in the runscript `am2/exp/fv/fms_runscript` accordingly.

## 3. Run the Model

### 3.1. Functionality of the Sample Runscripts

This release includes a compile script and a run script for the Finite-Volume AM2 model in the `am2/exp/fv` [`./exp/fv`] directory. The compile script

- compiles the `mppnccombine` executable, which combines individual files from multiprocessor output into one netcdf file,
- compiles and links the model source code,

The run script

- creates a working directory where the model will be run,
- creates or copies the required input data into the working directory, and
- runs the model.

Note that the directory paths and file paths are variables. They are initially set to correspond to the directory

structure as it exists upon download, but are made variables to accommodate changes to this directory structure.

Minimally, one should set the `root` directory to the location of the downloaded code. The `work` directory should be a temporary directory where the code will run. Note that a large amount of data will be copied into the `work` directory. Output from the run will be moved to the `outputDir` directory, which is customizable in the `runscript`.

For the single processor one day test case, the script took about 14 minutes to run on the SGI Altix Intel Itanium2 at GFDL with `ifort.9.0.027` and `mpt-1.12-1`.

The diagnostic fields output from the models is multithreaded. That is, each processor writes a separate file which includes data only from its own portion of the globe (its "domain"). A utility named `mppnccombine` is supplied which is executed after the model execution is complete and which combines these into a single file which covers the entire globe. For a complete description of `mppnccombine` see the [mppnccombine documentation](http://www.gfdl.noaa.gov/~fms/pubrel/j/atm_dycores/doc/dycore_public_manual.html#mppnccombine) [http://www.gfdl.noaa.gov/~fms/pubrel/j/atm\_dycores/doc/dycore\_public\_manual.html#mppnccombine] in the atmospheric dynamical core user's guide.

## 3.2. Portability Issues with the Sample Runscripts

If you encounter a compile error when executing the compile script, please first check whether you have correctly customized your `mkmf` template. The scripts use the `mkmf` utility, which creates a Makefile to facilitate compilation. The `mkmf` utility uses a platform-specific template for setting up system and platform dependent parameters. Sample templates for various platforms are provided in the `am2/bin` directory. You may need to consult your system administrator to set up a compilation template for your platform and ensure the locations for system libraries are defined correctly. For a complete description of `mkmf` see the [mkmf documentation](#) [../bin/mk-mf.html].

## 3.3. Changing the Sample Runscripts

### 3.3.1. Changing the length of the run and atmospheric time step

By default the scripts are set up to run only one day. The run length is controlled by the `namelist coupler_nml` which is set directly in the `runscript` for convenience.

### 3.3.2. Changing the number of processors

By default the scripts are set up to run with the MPI library, but only on one processor. To increase the number of processors, change the `$npes` variable at the top of the sample `runscript`. The processor count must be consistent with the processor layouts defined in `ice_model_nml`, `ocean_model_nml` and `fv_core_nml`.

To run without the MPI library, do the following:

1. Make sure you are only using one processor, ie, the variable `$npes` is set to 1 at the top of the sample `runscript`.
2. Change the run command in the `runscript` from `"mpirun -np $npes fms.x"` to simply `"fms.x"`.
3. Remove the `-Duse_libMPI` from the `mkmf` line in the compile script.
4. Remove the `-lmpi` from the `$LIBS` variable in your `mkmf` template.
5. Move or remove your previous compilation directory (specified as `$execdir` in the `runscript`) so that all code must be recompiled.

## 4. Examine the Output

You may download sample output data for comparison at <https://fms.gfdl.noaa.gov/projects/am2/> under the Files tab. The file `output.tar.gz` contains three directories: `ascii`, `history` and `restart`. The `ascii` directory contains text output of the model including `stdout` and log messages. The `history` directory contains netCDF diagnostic output, governed by your entries in the `diag_table`. History files are labeled with the timestamp at the beginning of the period of data they contain. The `restart` directory contains files which describe the state of the model at the time given in the timestamp; to start the model running from this state, you should provide a restart file as the initial conditions for your next run. The sample runscript manages restart files itself based on the `$numSubmissions` and `$runsPerScript` variables.