
WRF-GC

Release 2.0

The WRF-GC Authors

Jun 15, 2022

INTRODUCTION

1 Key concepts (Glossary)	3
2 Frequently Asked Questions	5
3 Getting started	11
4 Preparing your system	13
5 Downloading WRF-GC	17
6 Compiling WRF-GC	19
7 Domain, meteorology, and configuration	21
8 Chemical Initial/Boundary Conditions (IC/BC)	29
9 Running WRF-GC	33
10 Guide to outputs	35
11 Pre-flight checklist	37
12 Common errors	39
13 Nudging meteorology	45
14 Additional diagnostics	47
15 Adding species or updating chemical mechanism	49
16 Lightning NO_x emissions	53
17 Useful resources	55
Index	57

WRF-GC (WRF coupled with GEOS-Chem chemistry) is an online coupled model based on the regional meteorology model, WRF (Weather Research and Forecasting Model), and the GEOS-Chem chemical transport model.

This is the new web-based WRF-GC documentation. We are actively working on adding content.

Check out the [WRF-GC website](#) at the [Atmospheric Chemistry and Climate Group at SUSTech](#).

Note: This documentation is under active development. Some content may still be under the old documentation, with step-to-step instructions, [in this PDF](#).

The key references for WRF-GC are:

- **Lin et al., 2020:** Lin, H., Feng, X., Fu, T.-M., Tian, H., Ma, Y., Zhang, L., Jacob, D. J., Yantosca, R. M., Sulprizio, M. P., Lundgren, E. W., Zhuang, J., Zhang, Q., Lu, X., Zhang, L., Shen, L., Guo, J., Eastham, S. D., and Keller, C. A.: *WRF-GC (v1.0): online coupling of WRF (v3.9.1.1) and GEOS-Chem (v12.2.1) for regional atmospheric chemistry modeling – Part 1: Description of the one-way model*, *Geosci. Model Dev.*, 13, 3241–3265, <https://doi.org/10.5194/gmd-13-3241-2020>, 2020.
- **Feng et al., 2021:** Feng, X., Lin, H., Fu, T.-M., Sulprizio, M. P., Zhuang, J., Jacob, D. J., Tian, H., Ma, Y., Zhang, L., Wang, X., Chen, Q., and Han, Z.: *WRF-GC (v2.0): online two-way coupling of WRF (v3.9.1.1) and GEOS-Chem (v12.7.2) for modeling regional atmospheric chemistry–meteorology interactions*, *Geosci. Model Dev.*, 14, 3741–3768, <https://doi.org/10.5194/gmd-14-3741-2021>, 2021.

Most useful pages

- [Frequently Asked Questions](#) about WRF-GC.
- [Common errors](#) for when you run into errors. Ctrl+F (Command+F) will be your friend in this comprehensive page.
- The [Pre-flight checklist](#) has a list of all steps you have to go through for a successful WRF-GC run. Make sure you follow all steps.

Table of contents

KEY CONCEPTS (GLOSSARY)

Here is a list of concepts for WRF-GC that may be useful for using and developing WRF-GC.

WRF

The [Weather Research and Forecasting Model](#), a mesoscale weather model. It provides the meteorology and framework for WRF-GC to work.

GEOS-Chem

Atmospheric chemistry model that WRF-GC uses for chemistry. Not an acronym¹. [Website](#)

WRF-GC Model

An online, two-way coupling of WRF and GEOS-Chem into one model, using the **WRF-GC Coupler**. The two models do not run separately - WRF-GC runs through WRF and the coupler will run GEOS-Chem chemistry at the appropriate time steps. Running WRF-GC is very similar to running [WRF-Chem](#).

HEMCO

The Harmonized Emissions Component, an on-line emissions component that WRF-GC uses for emissions. WRF-GC uses a modified version of HEMCO for coupling to WRF-GC and is described by Lin et al., 2021².

WPS

The WRF Pre-Processor, a tool to prepare the grid and input meteorological fields (initial and boundary conditions) for the WRF model.

mozbc

A tool for inserting chemical initial and boundary conditions into WRF-Chem or WRF-GC's input files. We document the usage in [Chemical Initial/Boundary Conditions \(IC/BC\)](#).

¹ Cited from the GEOS-Chem narrative description: *The name "GEOS-Chem" was coined in 2001 and is first referred to in Bey et al. [2001]. It is not an acronym - there is nothing to spell out. GEOS stands for Goddard Earth Observing System and Chem stands for Chemistry but calling it the "Goddard Earth Observing System - Chemistry" model would be inappropriate because the GEOS Earth System Model can use other chemical modules besides GEOS-Chem, and GEOS-Chem can use other meteorological drivers besides GEOS.*

² Lin, H., Jacob, D. J., Lundgren, E. W., Sulprizio, M. P., Keller, C. A., Fritz, T. M., Eastham, S. D., Emmons, L. K., Campbell, P. C., Baker, B., Saylor, R. D., and Montuoro, R.: Harmonized Emissions Component (HEMCO) 3.0 as a versatile emissions component for atmospheric models: application in the GEOS-Chem, NASA GEOS, WRF-GC, CESM2, NOAA GEFS-Aerosol, and NOAA UFS models, *Geosci. Model Dev.*, 14, 5487–5506, <https://doi.org/10.5194/gmd-14-5487-2021>, 2021.

FREQUENTLY ASKED QUESTIONS

Note: This section is still heavily under construction as we aggregate questions and answers into this document.

2.1 Running environment

2.1.1 What are the system requirements for WRF-GC?

A Linux-based system, either on your local cluster or cloud services. Software requirements are: a Fortran/C compiler (Intel or GNU GCC+GFortran), MPI, HDF5, netCDF (with netCDF Fortran).

Recommended system specifications are at least 6 CPU cores, and at least 32 GB of RAM. Storage requirements depend on the resolution of your run, but plan for generally ~2 GB per output file. Input data for GEOS-Chem and WRF meteorology is also necessary, and may be around hundreds of GB.

2.1.2 Can I run WRF-GC on my own computer?

Generally we recommend running on a computing cluster. If this is not possible, cloud services such as AWS may also be available.

If you want to test WRF-GC locally, a Linux system with at least 8 *recent* (consumer CPUs: Intel Haswell, 4th-generation Core; AMD Ryzen) CPU cores, 32 GB of memory is recommended. However, running on your own computer may be extremely slow and only recommended for debugging.

2.1.3 I want to build a cluster for my lab to run WRF-GC. What do you recommend?

Hardware is ever-changing, but both recent AMD EPYC and Intel Xeon processors will work. If you are using an AMD system, you may want to use GNU compilers for better performance. Storage capacity is generally the more, the better, because WRF-GC outputs are fairly large as they include the 220+ species and all met fields computed by WRF and GEOS-Chem.

2.1.4 Does WRF-GC require ESMF?

No.

2.2 Building

2.2.1 Compiling WRF-GC takes a long time and is stuck on module_first_rk_step_part2.F90

This step usually takes approximately 40 minutes. Usually there is no cause for alarm unless compiling WRF-GC takes longer than 2-4 hours, depending on your machine.

2.3 Input

2.3.1 What are the typical input files for running WRF-GC?

Configuration:

- `namelist.input` - the configuration file (namelist) for WRF-GC. Most options are configured here.
- `HEMCO_Config.rc` - the HEMCO configuration file, for chemistry emissions.
- `input.geos` - only used to specify some input files paths for Fast-JX.

Input data:

- `met_em.*` files for meteorology boundary conditions: Generated by the WRF pre-processor system (WPS) based on reanalysis fields (e.g., GFS, FNL)
- `wrfinput_d01` and `wrfbdy_d01`: Initial and boundary condition files for WRF-GC (both meteorology and chemistry). Generated by `real.exe`, then chemical initial/boundary conditions are added to these files using a tool like `mozbc`. See *How do I generate chemical initial/boundary conditions?*
- Emissions inputs specified in paths within `HEMCO_Config.rc`.
- Essential chemistry input files specified within `input.geos`.

2.3.2 How do I generate chemical initial/boundary conditions?

See *Chemical Initial/Boundary Conditions (IC/BC)*.

2.3.3 Can I nudge the input meteorology data?

Yes. The process is similar to WRF, or WRF-Chem. More details soon.

2.3.4 Can I use other emission inventories?

Yes. You want to edit the HEMCO configuration file (`HEMCO_Config.rc`), like in GEOS-Chem. Generally, all emission inventories supported by GEOS-Chem/HEMCO are supported in WRF-GC, because the species list is the same.

For emission inventories from WRF-Chem (e.g., prepared by `prep_chem_sources`), these inventories need to be ported to HEMCO to function with WRF-GC. We do not support WRF-Chem style emissions (i.e., “auxinput05”) within WRF-GC.

2.4 Running and configuration/

2.4.1 Can I use WRF-GC to run Hg/CH4/CO2 specialty simulations?

Generally not out of the box. WRF-GC only supports chemical mechanisms for full-chemistry simulations. Xu et al. (2022) has developed a new version (WRF-GC-Hg v1.0) for atmospheric Hg. You can refer to [Xu et al., 2022 on GMD](#) for more information on the development of Hg simulation on WRF-GC.

2.4.2 Can I use WRF-GC with tropchem (troposphere-only chemistry)?

Generally not out of the box. By default, UCX will run above the tropopause, into the stratosphere. Note that WRF model top is generally lower than GEOS-Chem (which goes to 0.1 hPa). It is recommended to leave this at the default configuration - note that recent versions of GEOS-Chem have also retired tropchem.

2.4.3 Can I customize WRF-GC’s vertical grid?

Yes. See [WRF User’s Guide on vertical grid configuration](#). Specifically, `p_top_requested` in `namelist.input` sets the model top target (note that WPS must have the necessary meteorology up to this pressure level), and `e_vert` specifies the number of vertical levels requested.

If only `e_vert` and `p_top_requested` are specified, then `real.exe` will generate vertical levels for you and output some information. You may also be able to specify the grid completely manually using `eta_levels`.

2.4.4 Where are the configuration files for WRF-GC?

In the run directory. Generally, `WRFV3/run` or `WRF/run`. The configuration files you want to use are `namelist.input` (WRF namelist - configures both WRF and chemistry options), and `HEMCO_Config.rc` (to configure HEMCO emissions).

`input.geos` also holds the paths to some essential chemistry input files. You generally only need to edit the paths in this file.

Note: While `input.geos`, `HISTORY.rc`, and `HEMCO_Diagn.rc` files, familiar to GEOS-Chem users, are also in the WRF-GC run directory, they should generally not be modified. To control WRF-GC, use the WRF namelist `namelist.input`.

<p>Warning: Be careful to back up your configuration files. Every WRF-GC recompile will reset the namelist and configuration files.</p>

2.4.5 Can I run the model in multiple segmented runs?

Yes. WRF will generate restart files based on the namelist configuration's `restart_interval`.

2.4.6 What do we do about WRF parameterizations (e.g., cumulus) at higher resolution runs?

This is a research question, but the WRF-GC paper [Lin et al., 2020](#) includes *some* guidance:

The WRF-GC state conversion module currently supports convective mass flux calculations using the new Tiedtke scheme (Tiedtke, 1989; Zhang et al., 2011; Zhang and Wang, 2017) and the Zhang–McFarlane scheme (Zhang and McFarlane, 1995) (Table 1), because these two cumulus parameterization schemes are more physically compatible with the convective transport algorithm currently in GEOS-Chem. In addition, the users should consider the horizontal resolution of the model when choosing which cumulus parameterization to use. The new Tiedtke scheme and the Zhang–McFarlane schemes are generally recommended for use in simulations at horizontal resolutions larger than 10km (Skamarock et al., 2008; Arakawa and Jung, 2011). At horizontal resolutions between 2 and 10km, the so-called “convective grey zone” (Jeworrek et al., 2019), the use of the Grell–Freitas scheme is recommended for the WRF model (Grell and Freitas, 2014), as it allows subsidence to spread to neighboring columns; this option will be implemented in a future WRF-GC version. At horizontal resolutions finer than 2km, it is assumed that convections are resolved and cumulus parameterizations should not be used (Grell and Freitas, 2014; Jeworrek et al., 2019). The scale dependence of cumulus parameterizations and their impacts on convective mixing of chemical species are an active area of research, which we will explore in the future using WRF-GC.

2.5 Output

2.5.1 How can I configure output?

Use `history_interval` in WRF's `namelist.input`.

2.5.2 What is the output format? What are some tools to process them?

The output is in `wrfout_netCDF` format used by WRF, and WRF-Chem. As such, tools to process WRF and WRF-Chem outputs may be useful for WRF-GC with some species name modifications.

2.5.3 The outputs are so large! Can I compress them?

You may be able to use netCDF tools to only get the variables you want after output. If you want to customize how WRF-GC outputs variables, see [Can I select what variables WRF-GC outputs?](#).

2.5.4 Can I select what variables WRF-GC outputs?

Yes. [WRF documentation](#) has some instructions that make this possible. A text file (e.g., `outputlist.txt`) can be added to the run directory containing customized options, e.g.,

```
-.h:0:sala,salc
```

This will remove `sala` and `salc` from the output file (`wrfout_...`), when `namelist.input`'s `&time_control` section has the following:

```
iofields_filename = 'outputlist.txt',
```

2.5.5 Can I output GEOS-Chem diagnostics?

Limited support is available for this at this time. Generally, only very specific diagnostics such as wet deposition loss rates are available. See [Additional diagnostics](#) for detailed descriptions and guidance on how to manually output anything that is within GEOS-Chem / HEMCO.

Planeflight diagnostics are not available at this time but may be developed in the future.

2.6 Advanced

2.6.1 Does WRF-GC support MPI or OpenMP parallelization?

At present, only MPI. OpenMP routines were removed during the development of WRF-GC.

2.6.2 Which MPI does WRF-GC support?

`mvapich` was used for development, but `OpenMPI`, and `Intel MPI` should also work. When configuring WRF-GC you are asked to fill in the correct MPI library in the `ESMF_COMM` environment variable. `openmpi`, `mvapich2`, and `intelmpi` are supported.

If you have a different MPI communicator for compiling, you can try to edit these options with the correct linking flags for your MPI in `WRF/chem/Makefile`:

```
# Specify MPI-specific options (hplin, 6/23/19)
ifeq ($(ESMF_COMM),openmpi)
    MPI_OPT := $(shell mpif90 --showme:link)
    MPI_OPT += $(shell mpicxx --showme:link)
else ifeq ($(ESMF_COMM),mvapich2)
    MPI_OPT := -lmpich -lmpichf90
else ifeq ($(ESMF_COMM),intelmpi)
    MPI_OPT := -lmpi
else
    $(error Unknown MPI communicator ESMF_COMM, valid are openmpi or mvapich2)
endif
```

2.6.3 Does WRF-GC support parallel I/O by WRF?

Yes, but HEMCO does not use parallel I/O. You do not need PNETCDF to run WRF-GC normally.

GETTING STARTED

Still under construction. For now, please refer to the WRF-GC documentation for WRF-GC v2.0.

- [Running WRF with the GEOS-Chem Chemical Module \(User's Guide\), Updated for WRF-GC v2.0 \(PDF\)](#)
- [WRF-GC Website](#)

PREPARING YOUR SYSTEM

This section discusses:

- Required system **libraries** for running WRF-GC.
- Required **common input data** for running WRF-GC.

The steps in this page only need to be run once, and generally do not need to be re-done unless WRF-GC version updates.

4.1 Software requirements

You will need a **Linux** system with the following compilers, libraries, and tools:

- **Compiler:** Intel (version 15 or above), or gfortran.
- **MPI Library:** MVAPICH2 (version 2.3 or above), OpenMPI, or Intel MPI.
- zlib, hdf5 (version 1.8 or above), netCDF-C (version 4.6.1 or above), netCDF-Fortran (version 4.4.4 or above), JasPer (version 1.900)
- Git version management

If you already run GEOS-Chem, you may already have met the compiler and netCDF requirements.

Below are some specific guidance, but if you already know how to set up libraries, point WRF-GC to them in the environment configuration file, editing as necessary.

```
export CC=icc
export OMPI_CC=$CC

export CXX=icpc
export OMPI_CXX=$CXX

export FC=ifort
export F77=$FC
export F90=$FC
export OMPI_FC=$FC
export COMPILER=$FC
export ESMF_COMPILER=intel

# MPI Communication
export ESMF_COMM=openmpi
export MPI_ROOT=$MPI_HOME
```

(continues on next page)

(continued from previous page)

```
# Location of netCDF - SEE SPECIAL NOTE ABOUT NETCDF BELOW
export NETCDF=/n/holyscratch01/jacob_lab/hplin/wrfgc

# Location of JasPer library
export JASPERLIB=$JASPER_HOME/lib64
export JASPERINC=$JASPER_HOME/include
```

4.1.1 Running on a cluster

Your system administrator may have installed these packages. For example, on the Harvard “Cannon” cluster, the following libraries are used:

```
module load git/2.17.0-fasrc01
module load intel/17.0.4-fasrc01
module load openmpi/2.1.0-fasrc02
module load netcdf/4.5.0-fasrc02
module load netcdf-fortran/4.4.4-fasrc06
module load jasper/1.900.1-fasrc02
```

Note: Special note about the netCDF libraries. Since version 4, netCDF has been split into “netCDF-C” and “netCDF-Fortran”. **Both of these are required for WRF-GC**, and WRF expects them to be in the same folder. This is problematic in some clusters, as they may store `netcdf` and `netcdf-fortran` in different paths. The solution is to create your own paths, e.g., `~/wrf-gc/include`, `~/wrf-gc/bin`, and `~/wrf-gc/lib`, and link the netCDF-C and netCDF-Fortran files into these folders respectively. e.g.,

Then see the environment configuration above to point WRF-GC to the libraries.

4.1.2 Running on AWS or with Spack

Some of the package setup from [Haipeng’s guide on WRF-GC on AWS](#) may be helpful.

4.1.3 Running on your own system

Compile libraries in the following order:

- Install the compiler
- Install MPI (e.g., MVAPICH2)
- Install zlib
- Install JasPer
- Install HDF5
- Install netCDF-C
- Install netCDF-Fortran

When configuring these libraries, specify the install location somewhere, e.g.:

```
./configure --prefix=/home/example/wrf-gc
```

The general process to install libraries is to run `configure`, then `make`, then `make install`, then `make check` to make sure it is working.

Then see the environment configuration above to point WRF-GC to the libraries.

4.2 Data requirements

4.2.1 GEOS-Chem shared inputs (“ExtData”)

Note: If someone on your cluster / system already runs GEOS-Chem, you may have the data already. If so, you may be able to skip all of the below steps.

Download the GEOS-Chem input data directories, specifically HEMCO and CHEM_INPUTS. For HEMCO they are very large, and you only need to download folders corresponding to the simulation year(s) you expect to run.

Refer to the GEOS-Chem User’s Guide: * [Downloading data from WashU](#)

WRF-GC needs `ExtData/HEMCO/` and `ExtData/CHEM_INPUTS/`.

If you are running GEOS-Chem as well (e.g., for making WRF-GC boundary / initial conditions), you also need some meteorology fields, etc. Refer to the GEOS-Chem User’s Guide.

4.2.2 WRF Pre-Processor Geographical Data

Download the **required** geographical input data for WRF.

- For WRF version 3, visit [this page](#)
- For WRF version 4 and above, visit [this page](#)

For high-resolution simulations, we recommend downloading “Download Highest Resolution of each Mandatory Field”. This takes approximately ~50 GB of disk space.

4.3 Environmental configuration file (for reference)

Below is an example environment file using the Intel compilers + OpenMPI. The following need to be edited to fit your system:

- NETCDF needs to point to your netCDF install (inside this path there should be `bin`, `lib`, `include` for both NetCDF-C and NetCDF-Fortran)
- JASPERLIB needs to point to your JasPer install’s `lib` folder
- JASPERINC needs to point to your JasPer install’s `include` folder
- NETCDF_HOME and NETCDF_FORTRAN_HOME point to NetCDF-C and NetCDF-Fortran. Can be the same as `$NETCDF`

If you are using the GNU compilers (`gcc` and `gfortran`), you also need to edit:

- `CC=gcc, CXX=gcc, FC=gfortran, ESMF_COMPILER=gfortran`

If you are using other MPI libraries, you also need to edit:

- ESMF_COMM to openmpi, mvapich2, or intelmpi.
- If your system has weird, non-standard MPI installations, you may need to manually edit WRF-GC's chem/Makefile.

```
export CC=icc
export OMPI_CC=$CC

export CXX=icpc
export OMPI_CXX=$CXX

export FC=ifort
export F77=$FC
export F90=$FC
export OMPI_FC=$FC
export COMPILER=$FC
export ESMF_COMPILER=intel

# MPI Communication
export ESMF_COMM=openmpi
export MPI_ROOT=$MPI_HOME

export NETCDF=/n/holyscratch01/jacob_lab/hplin/wrfgc
export NETCDF_HOME=$NETCDF
export NETCDF_FORTRAN_HOME=$NETCDF
export JASPERLIB=$JASPER_HOME/lib64
export JASPERINC=$JASPER_HOME/include

# WRF options
export WRF_EM_CORE=1
export WRF_NMM_CORE=0
export WRF_CHEM=1

# needed forwrfv4
export NETCDF_classic=1

# GC extras (only for GC not WRF-GC)
export OMP_STACKSIZE=1000000000
export KMP_STACKSIZE=$OMP_STACKSIZE

# Base paths
export GC_BIN="$NETCDF_HOME/bin"
export GC_INCLUDE="$NETCDF_HOME/include"
export GC_LIB="$NETCDF_HOME/lib"

# If using NetCDF after the C/Fortran split (4.3+), then you will need to
# specify the following additional environment variables
export GC_F_BIN="$NETCDF_FORTRAN_HOME/bin"
export GC_F_INCLUDE="$NETCDF_FORTRAN_HOME/include"
export GC_F_LIB="$NETCDF_FORTRAN_HOME/lib"
```

DOWNLOADING WRF-GC

This section discusses:

- Downloading the *WRF* model.
- Downloading the WRF-GC coupler and the *GEOS-Chem* model.
- Downloading the *WPS* (WRF Pre-Processor).

Note: Currently, the GEOS-Chem model is included within the WRF-GC coupler. In the future, we plan to distribute these separately.

However, the GEOS-Chem model version within the WRF-GC coupler, located within the `gc` folder, **is unmodified**. You can change this copy of GEOS-Chem just as you would change any other copy of GEOS-Chem, for your research.

Choose a folder to store your WRF-GC model, then proceed with the steps below.

5.1 Downloading WRF

Obtain the WRF model from [the wrf-model/WRF GitHub](#)

The currently supported version is WRF v3.9.1.1.

Note: If you would like to use WRF version 4, please contact the WRF-GC team for an experimental version.

```
wget https://github.com/wrf-model/WRF/archive/refs/tags/V3.9.zip
unzip V3.9.zip
```

Extract WRF into the `WRFV3` folder (version 3) or `WRF` folder (version 4). For simplicity, we refer to these folders as `WRF` below.

5.2 Downloading the WRF-GC coupler and GEOS-Chem model

1. Go into the WRF folder and **delete** the existing WRF-Chem chemistry. Delete the chem folder entirely.
2. **Clone the WRF-GC repository into the chem folder.**

```
git clone https://github.com/jimmielin/wrf-gc-release.git chem
```

Make sure you have cloned this into the chem folder.

5.3 Downloading the WPS Pre-Processor

Go outside the WRF folder and download *WPS* side-by-side with WPS, from the [wrf-model/WPS GitHub](#). Download the version closest to your WRF version (3.9, 4.3, etc.)

Extract the WRF Pre-Processor.

Warning: If you are running simulations after July 2019, **you have to use WPS version 4.0 or above**. It works with WRF v3.9 as well. [Refer to this post for details](#)

COMPILING WRF-GC

This section discusses:

- Configuring WRF compile.
- **Installing the GEOS-Chem species into WRF.**
- Compiling WRF-GC.
- Compiling *WPS*.

6.1 Configuring WRF compile

Go to the WRF folder.

If you are using WRF version 3:

```
./configure -hyb
```

If you are using WRF version 4 or above:

```
./configure
```

Select the compiler option corresponding to your environment, and select the (**dmpar**) option. e.g., Intel compiler, `icc`, `ifort`, (`dmpar`). `dmpar` is required and means distributed-memory parallel, which uses MPI and is used by WRF-GC.

6.2 Installing the GEOS-Chem species into WRF

Warning: This step is **crucial**. If this step is not performed, you will have to start this page over and waste hours of your time.

Go to the WRF/chem folder and run `make install_registry`.

If this step shows an error, make sure you did download WRF-GC into WRF/chem folder. It should look like this:

```
WRF/chem$ ls
chem_driver.F      Makefile          gc                config
wrfgc_convert_state_mod.F ...
```

6.3 Compiling WRF-GC

Go back to the WRF folder and run:

```
./compile em_real
```

Wait patiently. The process may take up to 2 - 4 hours depending on your system. If you encounter errors, note them down. A list of common errors is available at [Common errors](#).

When successful, the output looks like

```
=====
build started:   Sun Apr 15 12:00:00 CST 2018
build completed: Sun Apr 15 13:00:00 CST 2018
--->                Executables successfully built                <---
-rwxrwxr-x 1 hplin hplin 63670720 Apr 15 05:34 main/ndown.exe
-rwxrwxr-x 1 hplin hplin 63753008 Apr 15 05:35 main/real.exe
-rwxrwxr-x 1 hplin hplin 62683008 Apr 15 05:34 main/tc.exe
-rwxrwxr-x 1 hplin hplin 71153896 Apr 15 05:34 main/wrf.exe
=====
```

6.4 Compiling WPS

1. Go one level up and then into the WPS folder.
2. Run `./configure`.
3. Run `./compile`.

If you encounter errors, note them down. A list of common errors is available at [Common errors](#).

DOMAIN, METEOROLOGY, AND CONFIGURATION

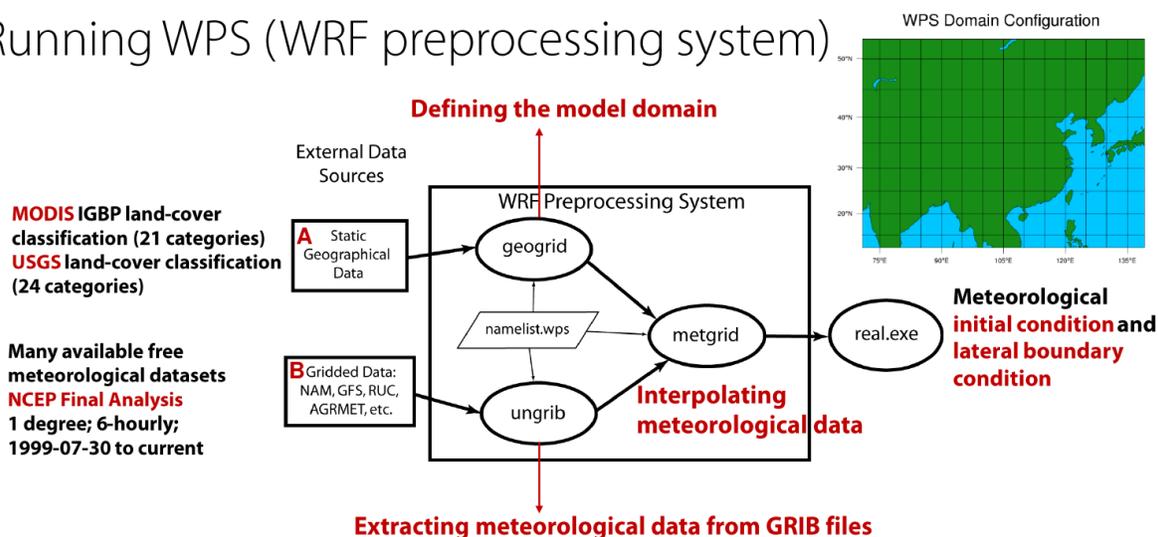
This section discusses:

- Setting up the domain using WPS's `geogrid.exe` tool.
- Processing the meteorological initial and boundary conditions by downloading, running WPS's `ungrib.exe`, and `metgrid.exe`.
- Basic configuration of WRF-GC using the `namelist.input`.
- Using WRF's `real.exe` to prepare input before adding in chemical initial/boundary conditions.

Note: The WRF Pre-Processor can be learned best from the [WRF User's Guide](#), as this is not specific to chemistry.

An overview of the workflow of the WRF Pre-Processor system (by Xu Feng):

Running WPS (WRF preprocessing system)



The data in "A" is described in *WRF Pre-Processor Geographical Data*. The data in "B" is described below in *Downloading meteorological data*.

7.1 Setting up the domain using GEOGRID

Configuration of the WPS is done in `namelist.wps` file under the WPS directory.

The first step is to describe your simulation domain. Example entries for the `namelist.wps` file's `&share` and `&geogrid` sections are below:

```
&share
wrf_core = 'ARW',
max_dom = 1,
start_date = '2016-06-27_00:00:00',
end_date   = '2016-06-29_00:00:00',
interval_seconds = 21600
io_form_geogrid = 2,
debug_level = 1,
/

&geogrid
parent_id      = 1,
parent_grid_ratio = 1,
i_parent_start = 1,
j_parent_start = 1,
e_we = 245,
e_sn = 181,

geog_data_res = 'default', 'default',
dx           = 27000,
dy           = 27000,
map_proj     = 'mercator',
ref_lat      = 27,
ref_lon      = 105,
truelat1     = 27.0,
stand_lon    = 105,
geog_data_path = '/n/seasasfs02/hplin/geog'
/
```

The configuration options you need to change with a brief description are listed below. This will get you up and running fast, but we recommend checking out the WPS user's guide.

- `max_dom`: Number of domains. 1 = single domain, up to 8 are supported when working with nested domains. We do not discuss multiple domains here for simplicity.
- `start_date` (per-domain): Start date of simulation
- `end_date` (per-domain): End date of simulation
- `e_we` and `e_sn`: Dimensions of the grid in x/y dimensions.
- `dx` and `dy`: Grid distance in the x/y dimensions where the map scale factor is 1. In meters when `map_proj = 'mercator'`, in degrees when `map_proj = 'lat-lon'`.
- `map_proj`: Map projection. **Only mercator and lat-lon (unrotated regular latitude-longitude) are supported currently in WRF-GC.**
- `ref_lat`, `ref_lon`, `truelat1`, `stand_lon` etc. are grid location parameters (where your regional grid is located in). Refer to the WRF User's Guide.
- `geog_data_path`: Path to the static WPS input data you downloaded in the previous steps.

Once `namelist.wps` is configured, you can run GEOGRID:

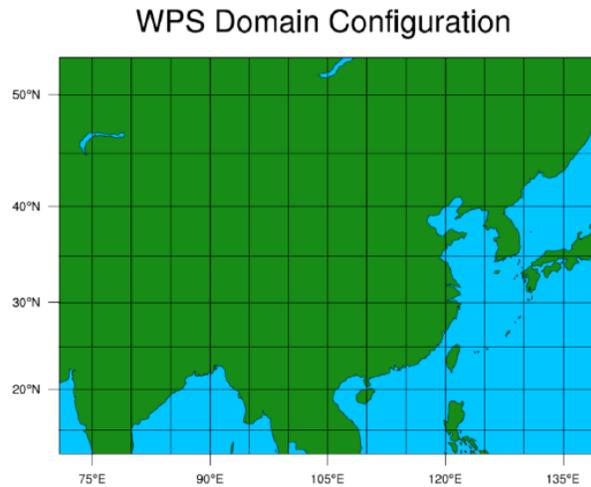
```
./geogrid.exe
```

This will generate `geo_em.d01.nc` (1 domain) and other `geo_em.d0X.nc` files for other domains if you are using multiple domains.

Preview the generated grid using the `nc1` script (requires NCL installed):

```
ncl util/plotgrids_new.ncl
```

An example is shown below:



7.2 Downloading meteorological data

Various types of meteorology can be used to drive WRF. We generally use [NCEP FNL](#) (Reanalysis) or [GFS](#) (Forecasts using GFS), but others may also be available. Refer to the WRF User's Guide.

7.3 Setting up Vtable

Depending on the meteorological data, the appropriate `Vtable` needs to be linked so the `UNGRIB` utility can find it.

If you are using [NCEP FNL](#) or [GFS](#) data, link the `Vtable.GFS` into WPS directory:

```
ln -s ungrib/Variable_Tables/Vtable.GFS Vtable
```

7.4 Running UNGRIB and METGRID

Configure UNGRIB and METGRID in `namelist.wps`. These should be mostly unchanged:

```
&ungrib
  out_format = 'WPS',
  prefix = 'FILE',
/

&metgrid
  fg_name = 'FILE',
  io_form_metgrid = 2,
/
```

Link GRIB files - `./link_grib.csh gfs*` (replace `gfs*` pointing to the meteorological input files you downloaded in the previous step)

Then run `./ungrib.exe`, then `./metgrid.exe`. You should now have meteorology data named `met_em.d...` in the WPS directory.

7.5 Link the meteorology from WPS to WRF

Go to the **WRF run directory** - `WRF/run`. Link the meteorological data into the run directory:

```
ln -sf ../../WPS/met_em* .
```

7.6 Configuring WRF-GC - `namelist.input`

Almost all WRF-GC configuration is performed inside `namelist.input`. This namelist, located in the WRF run directory, controls most aspects of the simulation.

Warning: Not all options in WRF for dynamics and physics are supported in WRF-GC! This is because to couple WRF to GEOS-Chem, the internal quantities need to be translated to GEOS-Chem's meteorology format (based on GEOS-FP).

The list of supported schemes is available in [Lin et al., 2020](#):

Namelist option	Description	Supported value
WPS		
max_dom	Maximum number of domains	1
map_proj	Map projection	lat-lon; mercator
geog_data_res	Static geographical data source	usgs_*
WRF dynamics		
hybrid_opt	Use hybrid sigma-pressure grid?	2 (Yes)
WRF physics		
bl_pbl_physics	Planetary boundary layer	All
cu_physics	Cumulus parameterization	7 (Zhang-McFarlane scheme), 16 (New Tiedtke scheme)
mp_physics	Microphysics option	6 (WRF single-moment six-class scheme), 8 (new Thompson scheme), 10 (Morrison double-moment scheme)
ra_lw_physics	Longwave radiation	3 (CAM3 scheme), 4 (RRTMG), 5 (new Goddard scheme)
ra_sw_physics	Shortwave radiation	4 (RRTMG shortwave)
sf_sfclay_physics	Surface layer	All
sf_surface_physics	Land surface	All
sf_lake_physics	Lake physics	All
sf_urban_physics	Urban surface	All

We do not discuss WRF configuration options in detail here and invite you to refer to the WRF User's Guide. The basic options to change in `namelist.input` are:

7.6.1 &time_control

- Configure the length of your run in `run_days/hours/minutes/seconds`, `start_year/month/day/hour/minute/second`, etc.
- **Configure output frequency.** Use `history_interval` (in minutes). e.g., hourly output - `history_interval = 60`.
- **Configure frames per output netCDF file.** e.g., `frames_per_outfile = 2` with `history_interval = 60` means 2 hours will be written per file.
- **Restarts.** If this is a restart run (running from existing `wrfrst_d<domain>_<date>` file), set `restart = .true.`. By default should be set to `.false.`.
- **Write out restart files.** Set `restart_interval` (in minutes).

7.6.2 &physics

- **Microphysics scheme.** (`mp_physics`): We recommend the Morrison Double-Moment scheme (`mp_physics = 10`).
- **Cumulus parameterization scheme.** (`cu_physics`): We recommend New-Tiedtke scheme (`cu_physics = 16`).

7.6.3 &chem

Configuration of chemistry is within the &chem section.

For WRF-GC chemistry, set `chem_opt = 233`.

You can control individual processes in GEOS-Chem using:

- Convection: `gc_do_convection`
- Emissions: `gc_do_hemco`
- Turbulence / Boundary layer mixing: `gc_do_pblmix`
- Chemistry: `gc_do_chemistry`
- Dry deposition: `gc_do_drydep`
- Wet deposition: `gc_do_wetdep`.

By setting these switches to 0 (off) or 1 (on).

To configure some simple GEOS-Chem diagnostics, add options to &chem following the guide in *Additional diagnostics*.

7.7 Configuring WRF-GC - `input.geos`

Most `input.geos` options known by GEOS-Chem users are not configured in `input.geos` in WRF-GC, and are instead controlled by `namelist.input`. Only two exceptions: the path to `CHEM_INPUTS` needs to be specified in:

```
Root data directory      : /n/holyscratch01/external_repos/GEOS-CHEM/gcgrid/data/ExtData/
```

and

```
%%% PHOTOLYSIS MENU %%% :  
FAST-JX directory       : /n/holyscratch01/external_repos/GEOS-CHEM/gcgrid/data/ExtData/  
↪CHEM_INPUTS/FAST_JX/v2021-10/
```

Most other options in `input.geos` for WRF-GC are ignored.

7.8 Configuring WRF-GC - emissions in `HEMCO_Config.rc`

Configuration of HEMCO is exactly the same as the GEOS-Chem model. Remember to update the HEMCO data path in this configuration file:

```
ROOT:                    /n/holyscratch01/external_repos/GEOS-CHEM/gcgrid/data/  
↪ExtData/HEMCO
```

Note: A reminder about `input.geos`, `HEMCO_Config.rc`, and `namelist.input` configuration files - **these files are replaced every time the WRF model is recompiled** (when `./compile_em_real` is ran). **Please remember to back up your configuration files!**

7.9 Running real.exe

After configuring, run `real.exe`. This is a memory and compute intensive operation - if you are on a cluster, you will need to submit a batch job like you would do when running other models. Otherwise, run

```
mpirun -np 32 ./real.exe
```

Where “32” would be the number of cores. The output can be watched by `tail -f rsl.out.0000` and any errors would be in `rsl.error.0000`.

After running `real.exe`, the initial condition file `wrfinput_d<domain>` and boundary condition file(s) `wrfbdy_d<domain>` are generated.

CHEMICAL INITIAL/BOUNDARY CONDITIONS (IC/BC)

WRF-GC is a regional model. Boundary conditions from a global model are generally required. Initial conditions can be provided by a WRF-GC run or other global models.

You can use MOZART4-GEOS5, CAM-chem, or GEOS-Chem as input for initial/boundary conditions.

The general process is as follows:

- Identify which data source you want to use as ic/bc.
- Set-up WRF-GC, and run `real.exe` to generate `wrfinput_d01` (initial conditions), and `wrfbdy_d01` (boundary conditions)
- Prepare ic/bc inputs into a netCDF format accepted by `mozbc`. ([mozbc for WRF-GC is located here](#))
- Use `mozbc` to add these inputs to `wrfinput_d01` and/or `wrfbdy_d01` files.
- Run the WRF-GC model.

8.1 Preparing IC/BC file from sources

8.1.1 Using global GEOS-Chem output as IC/BC

Details are available in the PDF getting started guide. [A MATLAB script](#) will process global GEOS-Chem outputs to netCDF files accepted by `mozbc`, which are then generated into WRF(-GC) initial/boundary files.

1. **Run the GEOS-Chem standard full-chemistry/tropchem simulation at a resolution of 2×2.5/4×5 degree (2×2.5 degree recommended).** The running time must cover the WRF-GC simulation period: e.g. if the simulation period of WRF-GC is from 2015-06-10 00:00:00 to 2015-06-20-00:00:00 (UTC), the time ranges for GEOS-Chem can be from 2015-06-07 00:00:00 to 2015-06-21 00:00:00 (UTC). Output the netCDF diagnostic files every 6 hours (00, 06, 12, 18), including

- (a) `GEOSChem.SpeciesConc.xxxxxxxxxxxxxx.nc4` (contains **instantaneous** `SpeciesConc_?ADV?`)
- (b) `GEOSChem.StateMet.xxxxxxxxxxxxxx.nc4` (contains `Met_PS1DRY`).

2. **Use the MATLAB script** `convert_gcoutput_mozart_structure_selected_domain.m` to merge the GEOS-Chem output files and reconstruct the data structure for `mozbc` to read.

Run the MATLAB script in the GEOS-Chem output file directory. Modify the script before running as follows:

- (a) `filename_input`: set the input filename as any one of the GEOS-Chem species concentration output files, e.g. `GEOSChem.SpeciesConc.20150601_0000z.nc4`.
- (b) `filename_output`: set the output filename freely.
- (c) `simulation_4_5/simulation_2_25`:

If the resolution of global GEOS-Chem simulation is 2×2.5 degree, please set it as follows:

```
simulation_4_5      = false;
simulation_2_25     = true;
```

If the resolution of global GEOS-Chem simulation is 4×5 degree, please set it as follows:

```
simulation_4_5      = true;
simulation_2_25     = false;
```

(d) Set the time ranges for output file

```
startyr             = 2015;
endyr               = 2015;
startmon            = 6;
endmon              = 6;
startdate           = 7;
enddate             = 21;
```

(e) Set the domain for output file (needs to be larger than your WRF-GC domain)

If the resolution of global GEOS-Chem simulation is 2x2.5:

longitude: 0 (index 1):2.5:357.5 (index 144) latitude: -90 (index 1):2:90 (index 91)

Here is an example:

```
lon_left            = 1; % longitude of western lateral condition
lon_right           = 73; % longitude of eastern lateral condition
lat_bottom          = 46; % latitude of southern lateral condition
lat_upper           = 91; % latitude of northern lateral condition
```

The netCDF file will be generated after running the script.

8.1.2 Using MOZART4-GEOS5/WACCM as IC/BC

Refer to the WRF-Chem documentation.

8.2 Using mozbc

8.2.1 Building mozbc (only need to do once)

1. **Download a customized version of mozbc for WRF-GC's hybrid sigma-eta grid.** This is available at Xu Feng's GitHub, [fengx7/mozbc_for_WRFv3.9](https://github.com/fengx7/mozbc_for_WRFv3.9)

```
git clone https://github.com/fengx7/mozbc_for_WRFv3.9.git mozbc
```

2. **Go to the mozbc directory downloaded and configure the environment.** Set up the path to your netCDF library

```
export NETCDF_DIR=/path/to/netcdf/here
```

The content inside \$NETCDF_DIR should have include, ... folders that correspond to netCDF.

3. **Compile mozbc.** Run `./make_mozbc`.

8.2.2 Configuring mozbc

Edit the input configuration file ending in `.inp`, corresponding to the version of GEOS-Chem you are using.

- GEOS-Chem 12.8.x: `GEOSCHEM_v12_8_1.inp`
- GEOS-Chem 12.2.1: `GEOSCHEM_v12_2_1.inp`

Configure the paths to the WRF input in `dir_wrf` (`wrfinput_d01`, `wrfbdy_d01` ... - run `real.exe` to generate these **first**) and the source netCDF file for IC/BC (created in step above). **Make sure the paths end in trailing slashes (/)**

```
dir_wrf = '/my/path/to/WRF/run/'
dir_moz = '/my/path/to/source/data/for/ic-bc/'
fn_moz  = 'wrfgc_icbc_data_test1.nc'
```

Run mozbc:

```
./mozbc < input.inp
```

Note: If you want to change the species mapping or add new species, please modify the `spc_map` in the input file, e.g.: `'isoprene -> ISOP'` where “isoprene” is the name of WRF-GC chemical species and “ISOP” is the name of GEOS-Chem species.

If the chemical IC/BC have been successfully written into the `wrfinput_d<domain>` and `wrfbdy_d<domain>` files, “bc_wrfchem completed successfully” will appear.

RUNNING WRF-GC

This section discusses:

- Running WRF-GC (`wrf.exe`)
- How to monitor progress.

This section assumes that all input data and configuration are ready. For configuration, refer to the previous sections.

9.1 Configuring WRF-GC

Refer to *Domain, meteorology, and configuration*'s relevant sections. Usually, you need to configure most options before running `real.exe`.

9.2 Running WRF-GC (`wrf.exe`)

Run it similarly to `real.exe`:

```
mpirun -np 32 ./wrf.exe
```

In a cluster you may want to use a batch script to do this. On a standalone Linux system you may want to do this within `screen`.

Track progress using `tail -f rsl.out.0000`. For troubleshooting, see the note at the top of *Common errors*.

GUIDE TO OUTPUTS

The WRF-GC outputs are provided by WRF in standard netCDF format.

Note: Are you a WRF or WRF-Chem user? Most of your tools for processing WRF-Chem output may only need species names to be changed to the GEOS-Chem names, and you will be good to go!

Here are some useful tools for analyzing WRF-GC outputs:

- **Python:** Standard tools for netCDF should work. There is also a [wrf-python](#) package for diagnostic and interpolation routines.
- **MATLAB:** You may find [m_map](#) useful for plotting.
- **NCL:** [The NCAR Command Language \(NCL\)](#) can read in NetCDF data and create plots based on many example scripts for WRF.

Your system may also have [ncview](#) which is a super handy tool to quickly look at netCDF files on your system. [Panoply](#) is useful as well.

10.1 Can I output GEOS-Chem / HEMCO diagnostics?

Unfortunately, not exactly as GEOS-Chem does. Please refer to [Additional diagnostics](#).

10.2 The outputs are too large / how can I compress them?

Please refer to the [Frequently Asked Questions](#)'s "Outputs" section.

PRE-FLIGHT CHECKLIST

Here is a list of all the steps that should be performed before running WRF-GC. This may be useful if you are familiar with the model or prefer to see an overview of the process. The specifics are in the PDF guide and also in other sections of this documentation.

11.1 Downloading WRF-GC and input data (do once)

- **Download the WRF model and the WPS pre-processor.**
- **Download WRF-GC.** Go into the WRF model and delete the chem folder, then download WRF-GC into WRF/chem folder.
- **Download the GEOS-Chem input data.** If you already run GEOS-Chem, remember the GEOS-Chem input data path. Otherwise, download [GEOS-Chem data directories](#). All CHEM_INPUTS are required, then choose inventories you need for *HEMCO*.
- **Download the WPS geographical input data.** See *Preparing your system*.

11.2 Compiling WRF-GC

Note: Your WRF root directory may be WRFV3 or WRF. We use WRF for consistency.

- **Configure your bash/shell environment.** It should have NETCDF, HDF5, the appropriate PATH with access to netCDF binaries, and specify the MPI type, etc.
- Go into WRF and configure to your environment. Run `./configure -hyb` (WRF version 3) or `./configure` (WRF version 4+). Choose your compiler and (`dmpar`) option corresponding to your compiler.
- Go into WRF/chem and run `make install_registry`. **This is very important. If you did not do this before the following steps, you have to start over from here.**
- Compile WRF-GC: `./compile em_real`.

Note: If you change the code **but not the species list**, then run `./compile em_real` to recompile directly.

If you change the species list, then `./clean -a` needs to be issued, then `./configure -hyb` from scratch and `./compile em_real`.

11.3 Compiling WPS

Compile WPS **after** you compile WRF-GC. `./configure`, `./compile`.

11.4 Running WRF-GC

- **Update configuration files to correct paths.** Paths to your GEOS-Chem input data are specified in **three locations**: `HEMCO_Config.rc`, and twice in `input.geos`.
- **Update `namelist.input` to be consistent with `namelist.wps`.**
- Link the meteorology data to your run directory (`ln -sf ../../WPS/met_em* .`)
- Run `real.exe` using MPI (e.g., `mpirun -np $SLURM_NTASKS ./real.exe` - depending on your cluster configuration you may need to batch this).
- **Create the initial/boundary conditions using `mozbc` or another tool.** Refer to *Chemical Initial/Boundary Conditions (IC/BC)*.
- **Run WRF-GC.** Run `wrf.exe`.

COMMON ERRORS

We hope you don't run into these errors. But if you do, we hope they will be a useful reference.

Tip: Use Ctrl+F (or Command+F) in your browser to search this page, it'll be much faster!

Note: WRF-GC has several output files. Most GEOS-Chem output is in the root CPU output file, `rsl.out.0000`. **Look in this file for any GEOS-Chem related errors.** HEMCO-related errors are in `HEMCO.log`. WRF-related errors can be in any of the files numbered `rsl.error.*` and `rsl.out.*`. You may want to use `tail -n 5 rsl.* | less` to quickly look at all log files to identify errors.

Warning: The HEMCO error log, `HEMCO.log`, only writes errors/warnings from the root process. In order to see HEMCO errors for all processes, edit `HEMCO_Config.rc` and set `LogFile` to `*`. Then you will be able to see HEMCO logs for all processes in the `rsl.out.*` files.

12.1 Compiling errors

12.1.1 error #7013: This module file was not generated by any release of this compiler.

Did you copy your WRF-GC directory from somewhere else? Some module files appear to be lingering over from a previous compile. Try running `./clean -a`, reconfiguring WRF-GC (`./configure -hyb` for WRFV3, `./configure` for WRFV4), and rebuilding (`./compile em_real`).

12.1.2 `ld: cannot find -lGIGC -lHistory -lGeosCore -lHistory -lKpp -lGeosCore -lIsorropia -lHCOI -lHCOX -lHCO -lGeosUtil -lKpp -lHeaders -lNcUtils`

A few things to check:

1. Go to `WRF/chem/gc/lib`. Do you see **all** these files?

```
WRF/chem/gc/lib$ ls
libGeosCore.a libGIGC.a libHCOI.a libHeaders.a libIsorropia.a libNcUtils.a
libGeosUtil.a libHCO.a libHCOX.a libHistory.a libKpp.a libObsPack.a
```

If yes, proceed to 2. If no, GEOS-Chem was not compiled correctly. Go to `WRF/chem`, run `make compile_chem`, then check if there are any errors.

2. Go to WRF/main/Makefile. Search for something like `chem/gc/lib: ... -L/n/home04/hplin/wrfgc_dev_202110/WRF/chem/gc/lib -lHistory ...`

Does this path, which points to `chem/gc/lib`, **exist** and **is correct**? Is it the same path as 1?

If no, go to step 3. **The error is because you (or someone) moved your WRF-GC directly after compiling it once. We can fix this.**

3. Move `WRF/main/Makefile.bak` to `WRF/main/Makefile`. Make sure this replaced version does not have `-lGIGC` **anywhere** within this file. If yes, download a fresh copy of WRF and move the `main/Makefile` from there into `WRF/main/Makefile`.
4. Now, go to `WRF/chem`, run `make compile_chem`. Verify that `WRF/chem/main/Makefile` has `chem/gc/lib` pointing to the right path. **Go back to the root WRF directory and compile again:** `./compile em_real`. No clean is necessary.

12.1.3 `mo_mozart_lib.f90:118: undefined reference to nf_open_`

This happens in `mozbc`. Check if you have **netCDF-C** and **netCDF-Fortran** in the same directory. Your `$NETCDF_DIR` should include `include/libnetcdf.a` - if not, you're missing the `netCDF-Fortran`.

12.1.4 `lib64: file not recognized: Is a directory; make[1]: [geogrid.exe] Error 1 (ignored)`

This may happen if you are on a cluster and are compiling `WPS`. Run `export MPI_LIB=""` then recompile `WPS`.

12.2 WPS-related errors

12.2.1 ERROR: The mandatory field TT was not found in any input data.

This is because GFS was updated in June 2019. Download and compile WPS version 4 (only WPS) and copy the `ungrib.exe` from WPS v4 and use that. [Haipeng has previously written on this](#)

12.3 WRF-related errors

12.3.1 ERROR: Mismatch between namelist and input file dimensions NOTE: 1 namelist vs input data inconsistencies found.

If you see an error similar to:

```
d01 2016-05-11_12:00:00 input_wrf.F: SIZE MISMATCH: namelist ide,jde,num_metgrid_
↪ levels=          245          181          27
d01 2016-05-11_12:00:00 input_wrf.F: SIZE MISMATCH: input file ide,jde,kde
↪ =              245          181          32
d01 2016-05-11_12:00:00 ---- ERROR: Mismatch between namelist and input file dimensions
NOTE:          1 namelist vs input data inconsistencies found.
```

You've hit a problem where your meteorology data source (e.g., GFS) has updated **during your model run times!** This doesn't happen often but you are bound to run into it sometime - see the list of updates and their dates [on this changelog page](#) from NOAA.

12.4.3 My NO_x / HNO₃ or nitrogen-related species are extremely high!

If this is happening near the coast, this is a bug with the ParaNO_x extension. This may even cause crashes in the model with negative species concentrations or KPP errors.

A temporary fix is to disable the ship plume extension (ParaNO_x). Go to `HEMCO_Config.rc` and turn off the *ParaNO_x* extension:

```
102      ParaNOx                : on      NO/NO2/O3/HNO3
```

Changing on to off. We are looking for a more permanent fix and are aware that this change may introduce differences in your simulation.

12.4.4 FAST-JX (RD_XXX): REQUIRED FILE NOT FOUND

Make sure you have updated `input.geos`'s configuration with the correct path to the Fast-JX CHEM_INPUTS:

```
%%% PHOTOLYSIS MENU %%% :
FAST-JX directory       : /n/holyscratch01/external_repos/GEOS-CHEM/gcgrid/data/ExtData/
↪CHEM_INPUTS/FAST_JX/v2019-10/
```

Make sure to update the path on top of `input.geos` as well:

```
Root data directory     : /n/holyscratch01/external_repos/GEOS-CHEM/gcgrid/data/ExtData/
```

Data can be obtained from [GEOS-Chem input data](#) at WUSTL.

12.5 HEMCO related errors

12.5.1 HEMCO ERROR: Invalid time index in (file)

Check the file that it is pointing to. There are a few likely reasons:

- The file was corrupted / not fully downloaded. Try opening it with `ncview` and checking!
- This inventory does not have the appropriate file for this date/time.

If (and only if) you have daily updating emissions data stored in monthly files and you are getting this error at the last day of the month, you are being affected by [this bug](#). There is a temporary fix for WRF-GC available, contact Haipeng Lin for details.

12.5.2 GEOS-Chem ERROR: Error encountered in “HCO_Run”! -> at HCOI_GC_Run (in module GeosCore/hcoi_gc_main_mod.F90)

This is a HEMCO error. Check `HEMCO.log`. If nothing is there, change `LogFile` to `*` in `HEMCO_Config.rc`, and check **all** `rs1.out.*` files after re-running the model.

12.6 mozbc related errors

12.6.1 mo_wrfchem_lib.o: ... undefined reference to nf_open_ / nf_inq_varid_ / nf_get_vara_real_ / nf_close_ / nf_def_var_

This error when compiling mozbc is usually because the path to netCDF library, NETCDF_DIR, is incorrect.

Make sure that your NETCDF_DIR is set to the root directory of your netCDF installation, where under \$NETCDF_DIR/lib you can find the appropriate netCDF library (libnetcdff.so).

12.7 Red herrings

If you see anything on this list, this is not the root cause - **this means that there's an error somewhere else!**

12.7.1 HEMCO ERROR: MaxNest too low, cannot enter GET_TIMEIDX (hco_read_std_mod.F90)

Any kind of error that says HEMCO ERROR: MaxNest too low means that there is an error somewhere above, in HEMCO.log. Check further!

12.7.2 forrtl: severe (408): fort: (2): Subscript #1 of the array LOC has value 11 which is greater than the upper bound of 10

This means that there is an error in HEMCO.log - check the HEMCO log instead! Maybe inventories are missing, etc.

12.7.3 forrtl: severe (408): fort: (2): Subscript #1 of the array ZPJ has value 1 which is greater than the upper bound of -1

Same! Check *rsl.error* files for **another error** - it may be hundreds of lines above (searching for GEOS-Chem ERROR: or REQUIRED FILE NOT FOUND helps).

It may be helpful to first check this: *FAST-JX (RD_XXX): REQUIRED FILE NOT FOUND* to make sure your input.geos paths are correct.

This error indicates that another error occurred somewhere upstream.

12.7.4 forrtl: error (78): process killed (SIGTERM)

Check if your system has enough memory. Also, check *rsl.out.0000* – this is usually an error within GEOS-Chem.

12.7.5 forrtl: severe (174): SIGSEGV, segmentation fault occurred

Check if your system has enough memory and try to run with `ulimit -c unlimited, ulimit -s unlimited`. This may also mean there is an error somewhere else upstream. Look further below for a “stack trace”, e.g.,

```
forrtl: severe (174): SIGSEGV, segmentation fault occurred
Image                PC                Routine           Line           Source
wrf.exe              xxxxxxxxxxxxxxxx  Unknown          Unknown        Unknown
libpthread-2.17.s   xxxxxxxxxxxxxxxx  Unknown          Unknown        Unknown
wrf.exe              xxxxxxxxxxxxxxxx  error_mod_mp_erro 437           error_mod.F90
wrf.exe              xxxxxxxxxxxxxxxx  flexchem_mod_mp_d 1057          flexchem_mod.F90
wrf.exe              xxxxxxxxxxxxxxxx  chemistry_mod_mp_ 299           chemistry_mod.F90
wrf.exe              xxxxxxxxxxxxxxxx  gigc_chunk_mod_mp 1277          gigc_chunk_mod.F90
```

Then look in this page for the error that corresponds to where the model has crashed.

12.7.6 Exit 152 mpirun -np ./wrf.exe

Check if your system / cluster job is allowing for enough memory. Also, check errors in other `rs1.*` files.

12.8 Not actually errors

12.8.1 Cannot find -IGCHPint

This is not an error, ignore. If you cannot successfully compile WRF-GC, there is an error above in the compile log.

12.8.2 cannot stat 'geos': No such file or directory

Also shows sometimes as `make[5]: [exe] Error 1 (ignored)`. This is not an error, ignore. If you cannot successfully compile WRF-GC, there is an error above in the compile log.

12.8.3 Trapping levels are weird and they should not be equal

This doesn't seem to be an issue. If your run was interrupted, there might be an error elsewhere.

12.8.4 47 more processes have sent help message help-mpi-btl-openib.txt / no device params found

This is usually not an issue. If your run stopped, check all the other run files.

NUDGING METEOROLOGY

This process is WRF-specific but may be useful for your WRF-GC runs. The namelist options described here are also described in the [WRF User's Guide](#)

13.1 Nudging using input FNL data

1. Add the following lines to your `namelist.input`, into a new section `&fdda` as shown below:

```
&fdda
grid_fdda           = 1, 1, 1,
gfdda_inname        = "wrffdda_d<domain>",
gfdda_interval_m    = 360, 360, 360,
gfdda_end_h         = 9999, 9999, 9999,
io_form_gfdda       = 2,
fgdt                = 0, 0, 0,
if_no_pbl_nudging_uv = 1, 1, 1,
if_no_pbl_nudging_t = 1, 1, 1,
if_no_pbl_nudging_q = 1, 1, 1,
if_zfac_uv          = 1, 1, 1,
k_zfac_uv           = 10, 10, 10,
if_zfac_t           = 0, 0, 10,
k_zfac_t            = 10, 10, 10,
if_zfac_q           = 0, 0, 0,
k_zfac_q            = 10, 10, 10,
guv                 = 0.0003, 0.0003, 0.0003,
gt                  = 0.0003, 0.0003, 0.0003,
gq                  = 0.0003, 0.0003, 0.0003,
if_ramping          = 0,
dtramp_min          = 60.0
/
```

To change whether winds, temperature, and/or water vapor are nudged you can change `if_no_pbl_nudging_uv` (t or q.)

2. **Re-run `real.exe`.** This will generate `wrffdda_d<domain>` files in your run directory.
3. **Run `wrf.exe`.**

13.2 Nudging using surface and upper-air observations

Coming soon.

ADDITIONAL DIAGNOSTICS

By default, WRF-GC outputs **all instantaneous species mixing ratios (ppmv)** at the output frequency (`history_interval` in `namelist.input`). However, other diagnostics provided by GEOS-Chem may be useful for your research. We document some methods to write GEOS-Chem diagnostics in WRF-GC.

Note: Because WRF-GC uses WRF as output, GEOS-Chem diagnostics (specified in `HISTORY.rc`) are unavailable in WRF-GC. But they can be written using some light code editing.

14.1 Limited support for GEOS-Chem diagnostics

The following GEOS-Chem diagnostics (from the [GEOS-Chem netCDF/History diagnostics list](#)) are supported in a limited way:

Table 1: Limited GEOS-Chem diagnostics supported

Diagnostic name	GEOS-Chem name	WRF-GC name (output)
Cloud convection flux	CloudConvFlux	cldcnvflx_n#
Budget: Emissions and Dry deposition	BudgetEmisDryDep	gcemisdrydep_<area>_n#
Budget: Boundary layer mixing	BudgetMixing	gcmixing_<area>_n#
Budget: Convection	BudgetConvection	gcconv_<area>_n#
Budget: Chemistry	BudgetChemistry	gcchem_<area>_n#
Budget: Wet deposition	BudgetWetDep	gcwetdep_<area>_n#
Fraction of soluble species lost in convective updrafts	WetLossConvFrac	wetlscnvfrc_n#
Loss of soluble species in convective updrafts	WetLossConv	wetlscnv_n#
Loss of soluble species in large-scale precipitation	WetLossLS	wetlossls_n#

- For <area> corresponding to the “Budget” diagnostic series, this can be `full` (full column), `trop` (troposphere), and `pbl` (boundary layer).
- For n#, **only four species are supported in WRF-GC**. The species IDs corresponding to the four species chosen for diagnostic output are specified in `namelist.input`:

```
gc_diagn_spc_n0      = 0,
gc_diagn_spc_n1      = 0,
gc_diagn_spc_n2      = 0,
gc_diagn_spc_n3      = 0,
```

The species list corresponding to their IDs can be obtained in the `rsl.out.0000` log by setting `namelist.input`'s `debug_level` to 5, and it will be output after the first time step (search for `%%%%% WRFGC_Convert_State_Mod`

Chemistry State Export %%%%%%%%%). (Remember to set the debug level back to a lower level or your model will run slow!)

14.2 Writing custom diagnostics from the GEOS-Chem/HEMCO models

Please refer to [Haipeng's website on outputting extra diagnostics](#) for now. We will migrate this page later.

ADDING SPECIES OR UPDATING CHEMICAL MECHANISM

To customize the chemical mechanism used in WRF-GC, there are three locations that need to be updated:

- **The GEOS-Chem model.** You may be most familiar with this part. You may need to add the desired species, reactions, and processes to GEOS-Chem.
- **The WRF-GC coupler.** The coupler needs to be aware of the new species, so it can perform advection and history output for them. Output is always through WRF.
- **WRF.** WRF needs to know the names of the species to setup input of `ic/bc` and output.

We outline the relevant steps below, assuming some familiarity with Fortran:

15.1 GEOS-Chem model

15.1.1 Only changing reactions

Simply edit `chem/gc/KPP/Standard/standard.eqn` as you would do within regular GEOS-Chem. You need to rebuild the chemical mechanism code using KPP. After that, simply recompile WRF-GC (`./compile_em_real` at the WRF root directory) and you are good to go.

As a reminder, the GEOS-Chem source code is unmodified and resides in `WRF/chem/gc`.

WRF-GC does not use `Tropchem`.

15.1.2 Adding additional code

WRF-GC is based upon GCHP technology by Long et al., 2015¹ and Eastham et al., 2018². The equivalent to `main.F` which most GEOS-Chem users are used to is `chem/gc/GCHP/gigc_chunk_mod.F90`. You may find most of the subroutine calls within the subroutine `GIGC_Chunk_Run`.

We recommend reading `main.F` and `gigc_chunk_mod.F90` and figuring out where your code should be positioned.

If you use this method, and no species are added, all you need to change is within GEOS-Chem and WRF-GC will work seamlessly. And your code will also work for GCHP now!

¹ Long, M. S., Yantosca, R., Nielsen, J. E., Keller, C. A., da Silva, A., Sulprizio, M. P., Pawson, S., and Jacob, D. J.: Development of a grid-independent GEOS-Chem chemical transport model (v9-02) as an atmospheric chemistry module for Earth system models, *Geosci. Model Dev.*, 8, 595–602, <https://doi.org/10.5194/gmd-8-595-2015>, 2015.

² Eastham, S. D., Long, M. S., Keller, C. A., Lundgren, E., Yantosca, R. M., Zhuang, J., Li, C., Lee, C. J., Yannetti, M., Auer, B. M., Clune, T. L., Kouatchou, J., Putman, W. M., Thompson, M. A., Trayanov, A. L., Molod, A. M., Martin, R. V., and Jacob, D. J.: GEOS-Chem High Performance (GCHP v11-02c): a next-generation implementation of the GEOS-Chem chemical transport model for massively parallel applications, *Geosci. Model Dev.*, 11, 2941–2953, <https://doi.org/10.5194/gmd-11-2941-2018>, 2018.

15.1.3 Adding species

Generally you need to update `standard.eqn` (KPP), `species_database_mod.F90` within GEOS-Chem.

15.2 WRF-GC coupler

The species need to be passed to and from WRF at every time step. There are three locations to edit in the WRF-GC coupler, located in `WRF/chem/wrfgc_convert_state_mod.F`.

1. **Species index declarations**, e.g.,

```
integer :: gi_a3o2, gi_acet, gi_acta ! ...
```

Add the species with a sensible name, e.g., `gi_newname`.

Go to `WRFGC_IdxSetup`:

```
gi_a3o2 = IND_('A3O2')
gi_acet = IND_('ACET')
gi_acta = IND_('ACTA')
```

Add the species you added with the GEOS-Chem name in quotes, e.g., `gi_newname = IND_('NEWNAME')`.

2. **Passing species from WRF to GEOS-Chem** after advection:

```
State_Chm%Species(II, JJ, k, gi_acet) = chem(i, k, j, p_acet) * 1.0e-6_fp
State_Chm%Species(II, JJ, k, gi_acta) = chem(i, k, j, p_acta) * 1.0e-6_fp
```

Add your species similarly.

Note: Do not worry about defining `p_acet`, `p_acta`, etc. - these are auto-generated by WRF. Remember the lowercase name of the species, this is the WRF name of the species you added (e.g., `newname`). We will use it in the next section.

3. **Passing species from GEOS-Chem to WRF** after chemistry:

```
chem(i, k, j, p_a3o2) = State_Chm%Species(II, JJ, k, gi_a3o2) * 1.0e+6_fp
chem(i, k, j, p_acet) = State_Chm%Species(II, JJ, k, gi_acet) * 1.0e+6_fp
```

Add your species similarly.

15.3 WRF registry

Let WRF know of this species's existence by editing `WRF/chem/registry.chem`. Search for:

```
state real a3o2      ikjftb chem      1      - i0{12}rhusdf=(bdy_
↪interp:dt) "a3o2"      "A3O2 mixing ratio"      "ppmv"
state real acet     ikjftb chem      1      - i0{12}rhusdf=(bdy_
↪interp:dt) "acet"     "ACETONE mixing ratio"      "ppmv"
```

Similarly add your species here. **Make sure you search `registry.chem` to make sure you have not defined a species twice.** A lot of names are reserved for compatibility with WRF-Chem code and we have no control over this. **If a name is already defined you can get away with reusing it directly.**

Every time you update the Registry, you need to completely recompile WRF-GC. There is no shortcut to this, so make sure your edit of the registry is correct or the compile process will crash very early. Steps:

- Update the WRF registry. In the chem directory, run: `make install_registry`.
- Clean all WRF-GC compiled code. In the WRF directory, run `./clean -a`.
- Reconfigure WRF. `./configure -hyb` (WRF version 3) or `./configure` (WRF version 4). This is as if you are reinstalling WRF-GC.
- Compile WRF-GC. `./compile em_real`.

<p>Warning: Be careful to back up your configuration files. Every WRF-GC recompile will reset the namelist and configuration files.</p>

LIGHTNING NOX EMISSIONS

Lightning NOx emissions are supported in WRF-GC v2.0 through HEMCO nad WRF lightning schemes.

To turn on the lightning NOx emissions in WRF-GC v2.0, you should turn on the lightning scheme in WRF-GC. The lightning NOx emissions are highly dependent on the flash rate simulated by WRF model.

Two lightning schemes are supported in WRF-GC:

- PR92 (Price and Rind, 1992) based on maximum w
- PR92 based on 20 dBz top

Please set the options in the configuration file (`namelist.input`) as follows:

```
&physics  
lightning_option           = 2,  
lightning_dt               = 120,  
lightning_start_seconds   = 600,  
cellcount_method          = 0,  
iccg_method               = 2,  
do_radar_ref              = 1,  
/  

```


USEFUL RESOURCES

- Original **WRF-GC Documentation** (Updated for v2.0). [PDF](#)
- **WRF (v3.9) User's Guide**: [Website](#)

17.1 Tools

- WRF-GC initial/boundary conditions generator using global GEOS-Chem output. [GitHub \(fengx7/WRF-GC-GCC_ICBC\)](#)
- *mozbc* tool for WRF-GC with support for hybrid sigma-pressure vertical grids. [GitHub](#)

INDEX

G

GEOS-Chem, 3

H

HEMCO, 3

M

mozbc, 3

W

WPS, 3

WRF, 3

WRF-GC Model, 3