

# Downloading and Building EnKF

Donald Stark

National Center for Atmospheric Research (NCAR)

The Developmental Testbed Center (DTC)

*Tuesday 17 February, 2015*

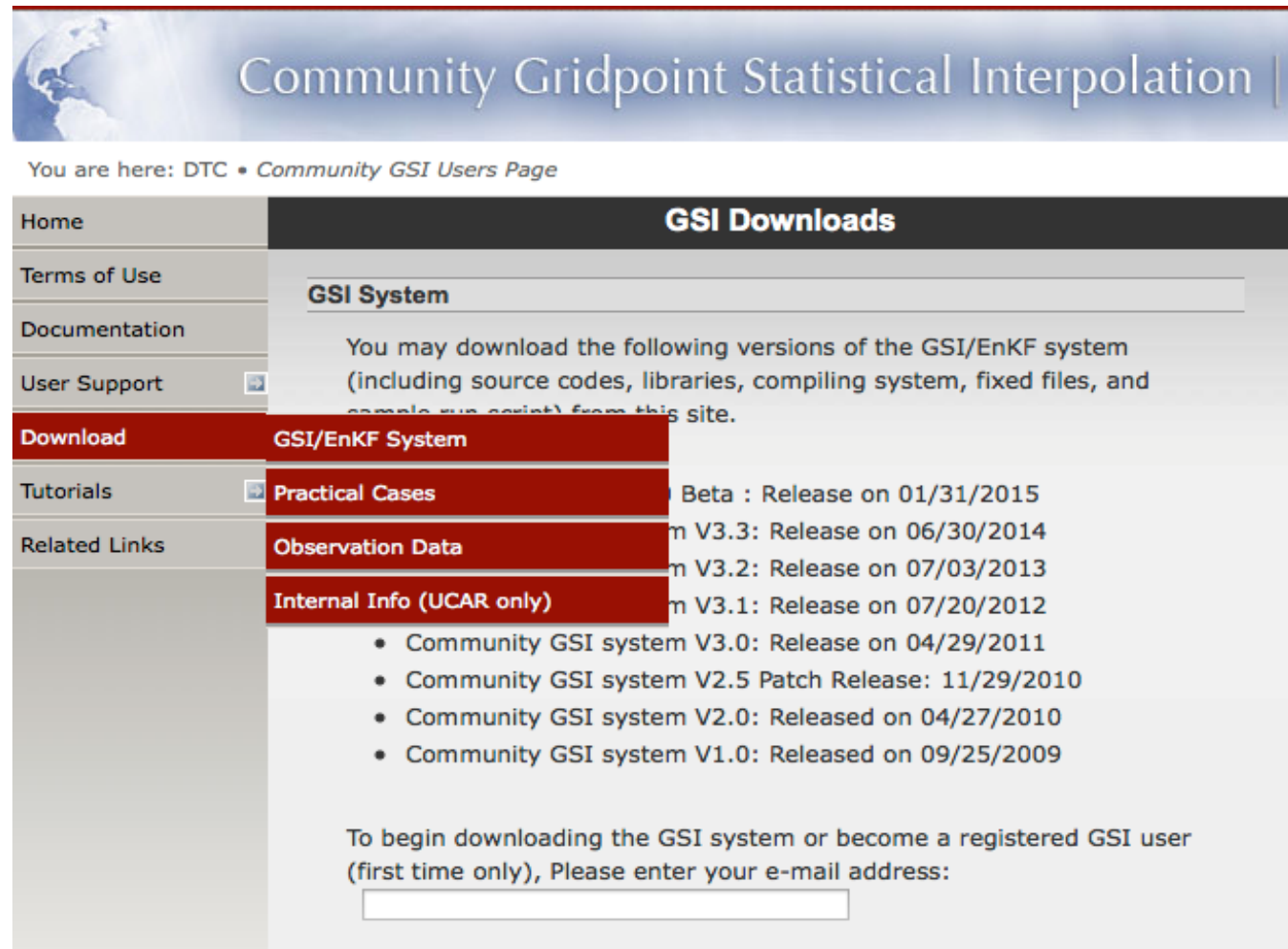
# Outline

- EnKF: Downloading and Building
  - Where to get the code
  - Directory structure
  - Unpacking, setup, & build
  - Porting build to new platforms

This talk follows the information contained with in Chapter 2 of the Community EnKF User's Guide for Release V1.0.

# Downloading the Source Code

- All of the GSI source code can be obtained from:
  - <http://www.dtcenter.org/com-GSI/users/downloads/index.php>



The screenshot shows the 'GSI Downloads' page. At the top, there is a header with a globe icon and the text 'Community Gridpoint Statistical Interpolation'. Below this is a breadcrumb trail: 'You are here: DTC • Community GSI Users Page'. The main content area is titled 'GSI Downloads' and contains a section for 'GSI System' with the following text: 'You may download the following versions of the GSI/EnKF system (including source codes, libraries, compiling system, fixed files, and sample run script) from this site.' Below this text is a list of download links, each with a corresponding release date: 'GSI/EnKF System Beta : Release on 01/31/2015', 'GSI/EnKF System V3.3: Release on 06/30/2014', 'GSI/EnKF System V3.2: Release on 07/03/2013', and 'GSI/EnKF System V3.1: Release on 07/20/2012'. Underneath these links is a bulleted list of previous releases: 'Community GSI system V3.0: Release on 04/29/2011', 'Community GSI system V2.5 Patch Release: 11/29/2010', 'Community GSI system V2.0: Released on 04/27/2010', and 'Community GSI system V1.0: Released on 09/25/2009'. At the bottom of the page, there is a text prompt: 'To begin downloading the GSI system or become a registered GSI user (first time only), Please enter your e-mail address:' followed by an empty text input field.

# Downloading Source code

## GSI Downloads

### Community EnKF Beta Version 1.0

Release of the community EnKF Beta Version 1.0 on January 31, 2015. This release pairs the community EnKF with version 3.3 of the community GSI release. The EnKF User's Guide is release with this version but the support will be available with the official release.

NOTE: This tarball includes the EnKF and GSI code, libraries, fixed files, run script, and utilities. CRTM coefficients are the same as the GSI release.

- [comGSI\\_EnKF\\_v3.3 \(93 MB\)](#)

Release notes [Check](#)

Known issues [Check](#)



# Unpack Downloads

- Source Code Tar file
  - comGSI\_EnKF\_v3.3.tar.gz
- Unpack source with
  - `tar -zxvf comGSI_EnKF_v3.3.tar.gz`
- Source Code directory
  - comGSI\_EnKF\_v3.3/
    - Includes both GSI and EnKF source code

# Tour of the Directory Structure

Inside the top level of the `comGSI_enkfv3.3/` directory are **four scripts** and **five directories**.

- `arch/`
- `clean`
- `compile`
- `configure`
- `fix/`
- `makefile`
- `run/`
- `src/`
- `util/`

# The rest

- **fix/** directory containing fixed parameter files
- **run/**
  - comgsi\_namelist.sh **namelist for GSI**
  - enkf\_wrf\_namelist.sh **namelist for EnKF**
  - run\_ensemble\_wrf.ksh **run script for EnKF analysis**
  - run\_gsi4enkf.ksh **run script to generate obs ensemble priors**
  - run\_gsi.ksh sample run script for regional analysis
  - run\_gsi\_global.ksh sample run script for global analysis
  - gsi.exe **executable**
- **src/** source directory
  - **libs/** **supplemental library source code**
  - **main/** **main GSI source code**
    - **Enkf/** **main EnKF source code**
- **util/** **additional community tools**

# Supplemental Libraries (libs/)

- **bacio/** NCEP BACIO library
- **bufr/** NCEP BUFR library
- **crtm\_2.1.3/** JCSDA Community Radiative Transfer Model
- **gsdcloud/** GSD Cloud Analysis
- **misc/** Misc additional libraries
- **nemsio/** Support for NEMS I/O
- **sfcio/** NCEP GFS surface file I/O module
- **sigio/** NCEP GFS atmospheric file I/O module
- **sp/** NCEP spectral-grid transforms (global application only)
- **w3emc\_v2.0.5/** NCEP W3 library (date/time manipulation, GRIB)
- **w3nco\_v2.0.6/** NCEP W3 library (date/time manipulation, GRIB)



# Building EnKF

---

# Building EnKF

- Build Sequence
  - Build WRF
  - Set library paths
    - `setenv WRF_DIR Location_of_WRF_directory`
    - `setenv LAPACK_PATH`
  - Build GSI
    - `./configure`
      - Customize file `configure.gsi` if necessary
    - `./compile`
    - Confirm executable resides in `run/` directory
      - `comGSI_EnKF_v3.3/run/gsi.exe`

# Building EnKF (continued)

- Build Sequence
  - Build EnKF
    - `cd src/main/enkf`
    - Run “make”
    - Makefile imports GSI `configure.gsi` file located in the top directory to define system variables
    - Uses object files in the `src/main/` directory
    - Confirm executable, default is called `wrf_enkf`, resides in the build directory

# EnKF build Versions

- Three possible EnKF Versions
  - Regional (Default, executable name `wrf_enkf`)
  - Global (executable name `global_enkf`)
  - NMMB (executable name `nmmb_enkf`)
- Version selection by editing file `src/main/enkf/Makefile.conf` around lines 45 to 50.

```
45 # FFLAGS_F90 = -DGFS
46 # EXE_FILE = global_enkf
47 FFLAGS_F90 = -DWRF
48 EXE_FILE = wrf_enkf
49 # FFLAGS_F90 = -DNMMB
50 # EXE_FILE = nmmb_enkf
```

# Diagnosing Build Issues

---

- System Requirements
- How the build system works
- What to do when the build fails

# System Requirements/Libraries

Same system requirements as GSI

- FORTRAN 90+ compiler
- C compiler
- Perl
- Gnu Make
- NetCDF V3.6+, & V4+
- Linear algebra library (ESSL, MKL or LAPACK/BLAS)
- MPI V1.2+ & OpenMP
- WRF V3.4.1+

# Supported Platforms/Compilers

	FC	CC
Intel	lfort 12.1.5, 13.0.1, 13.1.2, 14.0.2	lcc 12.1.5, 13.0.1, 13.1.2, 14.0.2
PGI	pgf90 12.10, 13.3, 13.9, 14.7, 14.9, 14.10	pgcc 12.10, 13.3, 13.9, 14.7, 14.9, 14.10

# Build Infrastructure

- Uses DTC Build system
- **/arch** directory contains rules & scripts for build.
  - **/arch/Config.pl** perl script for parsing system info & combining together *configure.gsi* file.
  - **/arch/preamble**: uniform requirements for the code, such as word size, etc.
  - **/arch/configure.defaults** default platform settings
  - **/arch/postamble**: standard make rules & dependencies
- **./clean** script to clean the build.
- **./configure** script to create configuration file *configure.gsi*; contains info on compiler, MPI, & paths.
- **./compile** script to compile executable.
- **./makefile** top level makefile for build.



## How the build works

- Running *./configure* creates file *configure.gsi* by:
  - Running the Perl script */arch/Config.pl*
  - Script *Config.pl* queries the system & selects the appropriate entry from */arch/configure.defaults*
  - Results are saved to *configure.gsi*.
  - Both GSI and EnKF builds use *configure.gsi*

# Identifying Build Errors

- Most build or run problems must be diagnosed by use of the log files.
- For build errors pipe the standard out and standard error into a log file with a command such as (for csh) `./compile |& tee build.log`
- Search the log file for any instance of the word "Error." Its presence indicates a build error. Be certain to use the exact spelling with a capital "E."
- If the build fails, but the word "Error" is not present in the log file, it typically indicates a failure in link the phase. Information on the failed linking phase will be present at the very end of the log file.

## configure.gsi

```
SHELL = /bin/sh
```

```
# Listing of options that are usually independent of machine type.
```

```
# When necessary, these are over-ridden by each architecture.
```

```
#### Architecture specific settings ####
```

```
# Settings for Linux x86_64, Intel compiler (ifort & icc) (dmpar,optimize)#
```

```
LDFLAGS = -Wl,-rpath,/glade/apps/opt/netcdf/4.3.0/intel/12.1.5/lib -openmp
```

```
COREDIR = /glade/scratch/stark/GSI/src/intel/release_V3.3_intel12.1.5
```

```
INC_DIR = $(COREDIR)/include
```

```
SFC = ifort
```

```
SF90 = ifort -free
```

```
SCC = icc
```

```
INC_FLAGS = -module $(INC_DIR) -I $(INC_DIR) -I /glade/apps/opt/netcdf/  
4.3.0/intel/12.1.5/include
```



## configure.gsi

```
SHELL = /bin/sh
```

```
# Listing of options that are usually independent of machine type.
```

```
# When necessary, these are over-ridden by each architecture.
```

```
#### Architecture specific settings ####
```

```
# Settings for Linux x86_64, Intel compiler (ifort & icc) (dmpar,optimize)#
```

```
LDFLAGS = -Wl,-rpath,/glade/apps/opt/netcdf/4.3.0/intel/12.1.5/lib -openmp
```

```
COREDIR = /glade/scratch/stark/GSI/src/intel/release_V3.3_intel12.1.5
```

```
INC_DIR = $(COREDIR)/include
```

```
SFC = ifort
```

```
SF90 = ifort -free
```

```
SCC = icc
```

```
INC_FLAGS = -module $(INC_DIR) -I $(INC_DIR) -I /glade/apps/opt/netcdf/  
4.3.0/intel/12.1.5/include
```



# configure.gsi

```
SHELL = /bin/sh
```

```
# Listing of options that are usually independent of machine type.
```

```
# When necessary, these are over-ridden by each architecture.
```

```
#### Architecture specific settings ####
```

```
# Settings for Linux x86_64, Intel compiler (ifort & icc) (dmpar,optimize)#
```

```
LDFLAGS = -Wl,-rpath,/glade/apps/opt/netcdf/4.3.0/intel/12.1.5/lib -openmp
```

```
COREDIR = /glade/scratch/stark/GSI/src/intel/release_V3.3_intel12.1.5
```

```
INC_DIR = $(COREDIR)/include
```

```
SFC = ifort
```

```
SF90 = ifort -free
```

```
SCC = icc
```

```
INC_FLAGS = -module $(INC_DIR) -I $(INC_DIR) -I /glade/apps/opt/netcdf/  
4.3.0/intel/12.1.5/include
```



# configure.gsi

```
SHELL = /bin/sh
```

```
# Listing of options that are usually independent of machine type.
```

```
# When necessary, these are over-ridden by each architecture.
```

```
#### Architecture specific settings ####
```

```
# Settings for Linux x86_64, GNU compilers (gfortran & gcc) (dmpar,optimize)#
```

```
LDFLAGS = -Wl,-noinherit-exec
```

```
COREDIR = /d1/stark/GSI/src/intel/V3.2/release_V3.2
```

```
INC_DIR = $(COREDIR)/include
```

```
SFC = ifort
```

```
SF90 = ifort -free
```

```
SCC = icc
```

```
INC_FLAGS = -I $(INC_DIR) -I /usr/local/netcdf3-ifort/include
```



## ***Fortran Build Flags:***

`FFLAGS_i4r4 = -integer-size 32 -real-size 32`

`FFLAGS_i4r8 = -integer-size 32 -real-size 64`

`FFLAGS_i8r8 = -integer-size 64 -real-size 64`

`FFLAGS_DEFAULT = -fp-model precise -assume byterecl -fpe0 -ftz -convert  
big_endian`

`FFLAGS_DEBUG = -O0 -g -traceback -check bounds -warn errors -fpstkchk -mp`

`FFLAGS_OPT = -O3`

`FFLAGS = -O3 $(FFLAGS_DEFAULT) $(INC_FLAGS) $(LD_FLAGS) -DLINUX`

# configure.gsi

# Library build flags

FFLAGS\_BACIO = -O3 \$(FFLAGS\_DEFAULT)

ARFLAGS\_BACIO =

FFLAGS\_BUFR = -O3 \$(FFLAGS\_DEFAULT) \$(FFLAGS\_i4r8)

CFLAGS\_BUFR = -O3 -DUNDERScore

ARFLAGS\_BUFR =

FFLAGS\_CLOUD = -O3 \$(FFLAGS\_DEFAULT)

FFLAGS\_CRTM = -O2 \$(FFLAGS\_DEFAULT)

LFLAGS\_CRTM =

FFLAGS\_GFSIO = -O3 \$(FFLAGS\_DEFAULT) \$(FFLAGS\_i4r4)

ARFLAGS\_GFSIO =

FFLAGS\_SFCIO = -O3 \$(FFLAGS\_DEFAULT) \$(FFLAGS\_i4r4)

FFLAGS\_SIGIO = -O3 \$(FFLAGS\_DEFAULT) \$(FFLAGS\_i4r4)

ARFLAGS\_SIGIO =

FFLAGS\_SP = -O3 \$(FFLAGS\_DEFAULT) \$(FFLAGS\_i4r8)

FFLAGS\_W3 = -O3 \$(FFLAGS\_DEFAULT)





# Getting Help

- For more detailed information on GSI installation see: Chapter 2 of the GSI User's Guide.
  - [www.dtcenter.org/com-GSI/users/docs/index.php](http://www.dtcenter.org/com-GSI/users/docs/index.php)
- For more detailed information on EnKF installation see: Chapter 2 of the EnKF User's Guide.
  - [www.dtcenter.org/com-GSI/users/docs/index.php](http://www.dtcenter.org/com-GSI/users/docs/index.php)
- Check the FAQ
  - [www.dtcenter.org/com-GSI/users/support/faqs/index.php](http://www.dtcenter.org/com-GSI/users/support/faqs/index.php)
- Check the Known Issues
  - [www.dtcenter.org/com-GSI/users/support/known\\_issues/index\\_v3.3.php](http://www.dtcenter.org/com-GSI/users/support/known_issues/index_v3.3.php)
- For further assistance contact:
  - [gsi\\_help@ucar.edu](mailto:gsi_help@ucar.edu)

