2012 GSI Summer Tutorial, Boulder, CO

GSI Fundamentals (1): Setup and Compilation

Donald Stark

National Center for Atmospheric Research (NCAR)

The Developmental Testbed Center (DTC)

Wednesday 21 August, 2012

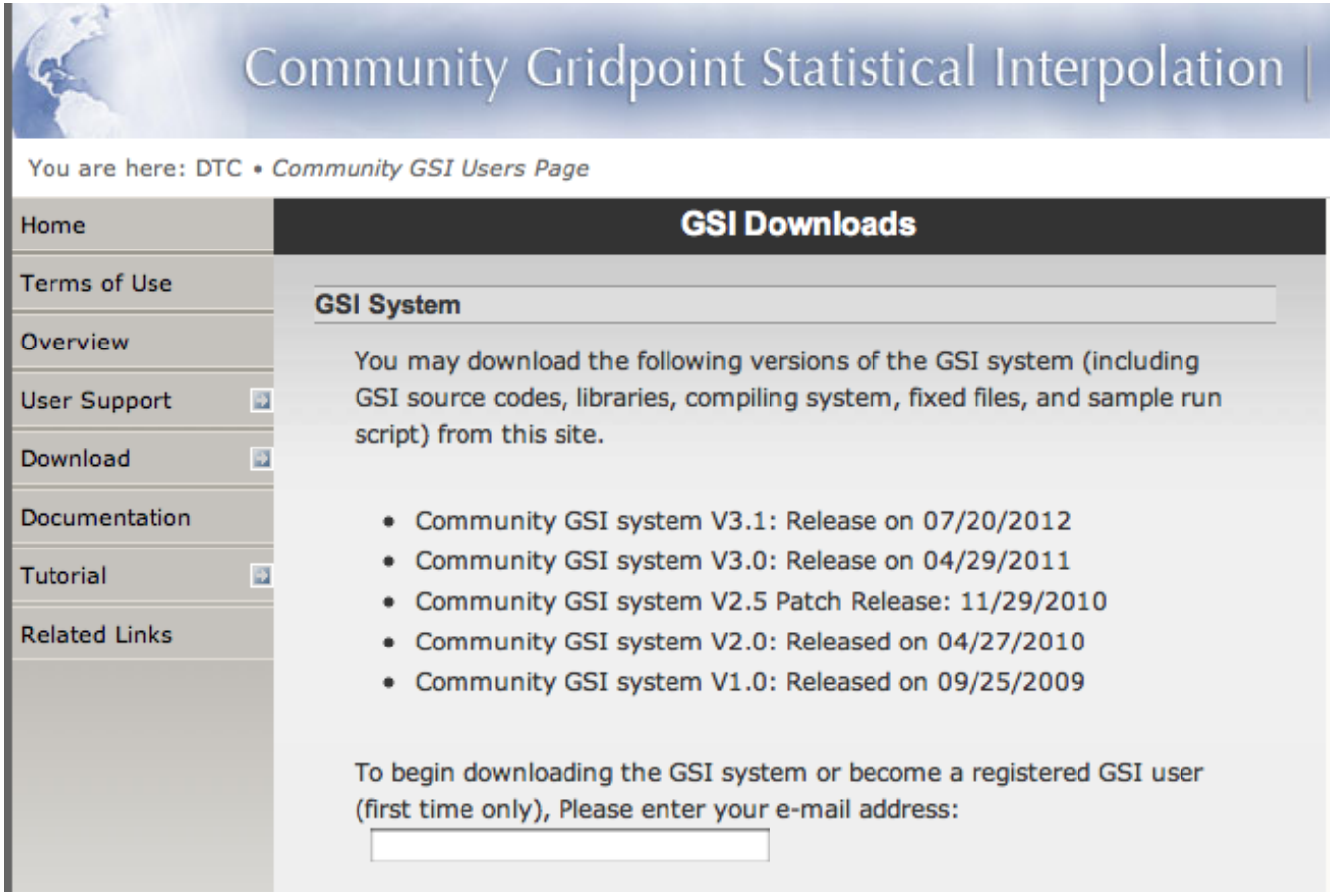
Outline

- GSI fundamentals (1): Setup and Compilation
  - Where to get the code
  - Directory structure
  - Unpacking, setup, & build
  - Porting build to new platforms
- GSI fundamentals (2): Run and Namelist
- GSI fundamentals (3): Diagnostics
- GSI fundamentals (4): Applications

This talk is tailored based on Chapter 2 of the GSI User's Guide for Community Release V3.1

Downloading the Source Code

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



Downloading Source code

GSI Downloads

Community GSI System Version 3.1

Official release of the community GSI Version 3.1 on July 20, 2012. Full documentation and support are available with the release.

NOTE: This tarball includes the GSI code, libraries, fixed files, run script, and utilities. It does not include CRTM cofficients. The CRTM 2.0.5 cofficients are available as a separate download. Both tarballs are necessary to run GSI.

- comGSI_v3.1 tarball (7.9 MB)
- CRTM 2.0.5 coefficients tarball (1.6 GB)

Release notes Check
Known issues Check

Unpack Downloads

- Two tar files
  - comGSI_v3.1.tar.gz
  - CRTM_Coefficients-2.0.5.tar.gz
- Unpack source code & CRTM coefficients
  - gunzip *.tar.gz
  - tar –xvf comGSI_v3.1.tar
  - tar –xvf CRTM_Coefficients-2.0.5.tar

Supported Platforms

Platform	F compiler	C compiler
IBM	xlf	xlc
Linux	Intel (ifort)	Intel (icc)
	Intel (ifort)	Gnu (gcc)
	PGI (pgf90)	PGI (pgcc)
	PGI (pgf90)	Gnu (gcc)
Mac Darwin	PGI (pgf90)	PGI (pgcc)

System Requirements

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

Tour of the Directory Structure

Inside the top level of the comGSI_v3.1/ directory are four scripts and five directories.

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

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 infomation on compiler, MPI, & paths.
- **./compile** script to compile executable.
- **./makefile** top level makefile for build.

The rest

- **fix/** directory containing fixed files
  - Background error covariance and observation errors
  - CRTM coefficients – moved to a separate directory due to size
  - Observation data control files
  - BUFR tables for BUFR/PrepBUFR files
- **run/**
  - run_gsi.ksh sample GSI run script
  - run_gsi_angupdate.ksh sample satellite angle bias correction run script
  - gsi.exe executable after compiling
- **src/** source directory
  - **libs/** supplemental library source code
  - **main/** main GSI source code
- **util/** additional GSI community tools

Supplemental Libraries (libs/)

- **bacio/** NCEP BACIO library
- **bufr/** NCEP BUFR library
- **crtm_jcsda_2.0/** JCSDA Community Radiative Transfer Model
- **gfsio/** Unformatted Fortran record for GFS I/O
- **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)
- **w3/** NCEP W3 library (date/time manipulation, GRIB)

Building GSI

Building GSI

- Build sequence
  - ./clean –a
  - Set library paths
    - setenv WRF_DIR *Location_of_WRF_directory*
    - setenv LAPACK (*typically only needed for Linux w/Intel*).
  - ./configure
    - Customize file *configure.gsi* if necessary
  - ./compile >& compile.log &

- Successful compilation
  - *comGSI_v3.1/run/gsi.exe*
  - *No "Error" information in "compile.log"*

Clean Compilation

- To remove all object files and executables, type:

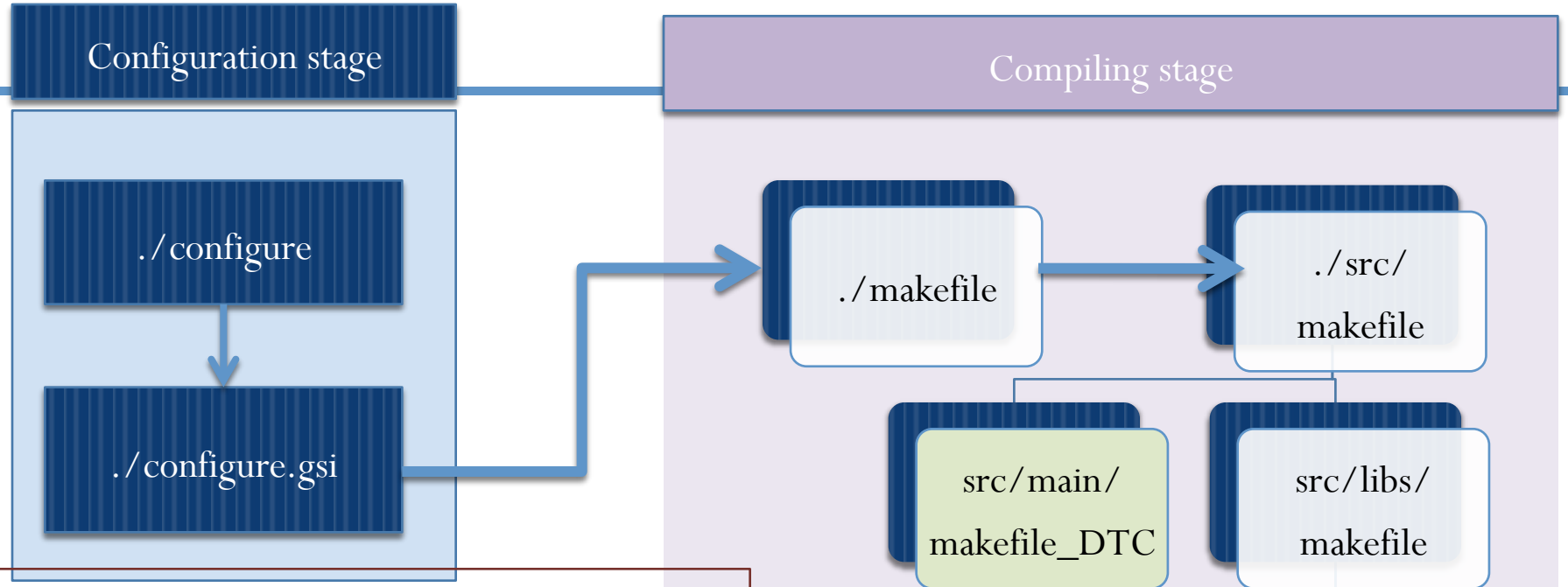
  clean

- To remove **all** built files, including the configure file, type: *clean –a*
  - A clean all needed if:
    - Compilation failed
    - Want to change configuration file

Diagnosing Build Issues

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

GSI Build System



- ✓ Run ./configure
  - Generate local compile options
  - Saves values in configure.gsi
- ✓ Run ./compile
  - Runs top level ./makefile
  - Which in turn runs ./src/makefile,
  - Which runs ./src/main/makefile

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.

Fixing Build Issues

- Most build problems are due to non-standard instillation of one of the following:
  - compiler,
  - MPI,
  - or support libraries.
- Edit paths in the file configure.gsi to correctly reflect your system.
- When build succeeds, modify file arch/configure.defaults to include new settings.
- Report issues to gsi_help@ucar.edu so they can be addressed in next release.

Fixing Build Issues (continued)

- The name or location of your LAPACK library may differ from what the build assumes. See **MYLIBsys**
- You may also want to use different Fortran compiler flags: See **FFLAGS_***
- You may also want to use different C compiler flags: See **CFLAGS**
- You may have a slightly different name for your compilers: See **SFC**, **SF90**, and **SCC** to specify your Fortran, Fortran90+, and C compilers.
- See the User's Guide for details

configure.gsi

```
#### Architecture specific settings ####
# Settings for Linux x86_64, PGI compilers (pgf90 & gcc)  (dmpar,optimize)#
LDFLAGS       =  -Wl,-noinhibit-exec

COREDIR       =  /d1/stark/GSI/src/pgi/release_V3.1
INC_DIR       =  $(COREDIR)/include
BYTE_ORDER    =  LITTLE_ENDIAN
SFC           =  pgf90
SF90          =  pgf90 -Mfree
SCC           =  gcc
INC_FLAGS     =  -module $(INC_DIR) -I $(INC_DIR) -I /usr/local/netcdf3-pgi/include
FFLAGS_i4r4   =  -i4 -r4
FFLAGS_i4r8   =  -i4 -r8
FFLAGS_i8r8   =  -i8 -r8
FFLAGS_DEFAULT =  -Kieee -pc 64 -Ktrap=fp
FFLAGS_DEBUG  =  -O0 -q -C
FFLAGS_OPT    =  -O3
FFLAGS        =  -fast $(FFLAGS_DEFAULT) -DLANGUAGE_FORTRAN -DsysLinux $(INC_FLAGS) $(LDFLAGS) -DLINUX -DPGI
# Library build flags
FFLAGS_BACIO  =  -O3 $(FFLAGS_DEFAULT)
ARFLAGS_BACIO =
FFLAGS_BUFR   =  -O3 $(FFLAGS_DEFAULT) $(FFLAGS_i4r8)
CFLAGS_BUFR   =  -O3 -DUNDERSCORE
FFLAGS_CLOUD  =  -O3 $(FFLAGS_DEFAULT)
FFLAGS_CRTM   =  -fast $(FFLAGS_DEFAULT)
LFLAGS_CRTM   =
FFLAGS_GFSIO  =  -O3 $(FFLAGS_DEFAULT) $(FFLAGS_i4r4)
ARFLAGS_GFSIO =
```

configure.gsi

```
#### Architecture specific settings ####
# Settings for Linux x86_64, PGI compilers (pgf90 & gcc)  (dmpar,optimize)#
LDFLAGS       =  -Wl,-noinhibit-exec

COREDIR       =  /d1/stark/GSI/src/pgi/release_V3.1
INC_DIR       =  $(COREDIR)/include
BYTE_ORDER    =  LITTLE_ENDIAN
SFC           =  pgf90
SF90          =  pgf90 -Mfree
SCC           =  gcc
INC_FLAGS     =  -module $(INC_DIR) -I $(INC_DIR) -I /usr/local/netcdf3-pgi/include
FFLAGS_i4r4   =  -i4 -r4
FFLAGS_i4r8   =  -i4 -r8
FFLAGS_i8r8   =  -i8 -r8
FFLAGS_DEFAULT =  -Kieee -pc 64 -Ktrap=fp
FFLAGS_DEBUG  =  -O0 -q -C
FFLAGS_OPT    =  -O3
FFLAGS        =  -fast $(FFLAGS_DEFAULT) -DLANGUAGE_FORTRAN -DsysLinux $(INC_FLAGS) $(LDFLAGS) -DLINUX -DPGI
# Library build flags
FFLAGS_BACIO  =  -O3 $(FFLAGS_DEFAULT)
ARFLAGS_BACIO =
FFLAGS_BUFR   =  -O3 $(FFLAGS_DEFAULT) $(FFLAGS_i4r8)
CFLAGS_BUFR   =  -O3 -DUNDERSCORE
FFLAGS_CLOUD  =  -O3 $(FFLAGS_DEFAULT)
FFLAGS_CRTM   =  -fast $(FFLAGS_DEFAULT)
LFLAGS_CRTM   =
FFLAGS_GFSIO  =  -O3 $(FFLAGS_DEFAULT) $(FFLAGS_i4r4)
ARFLAGS_GFSIO =
```

Paths & FFLAGS

configure.gsi

```
COREDIR      =  /d1/stark/GSI/src/pgi/release_V3.1
INC_DIR      =  $(COREDIR)/include
BYTE_ORDER   =  LITTLE_ENDIAN
SFC          =  pgf90
SF90         =  pgf90 -Mfree
SCC          =  gcc
INC_FLAGS    =  -module $(INC_DIR) -I $(INC_DIR) -I /
usr/local/netcdf3-pgi/include
```

Paths & FFLAGS

configure.gsi

```
COREDIR       =  /d1/stark/GSI/src/pgi/release_V3.1
INC_DIR       =  $(COREDIR)/include
BYTE_ORDER    =  LITTLE_ENDIAN
SFC           =  pgf90
SF90          =  pgf90 -Mfree
SCC           =  gcc
INC_FLAGS     =  -module $(INC_DIR) -I $(INC_DIR) -I /
usr/local/netcdf3-pgi/include
```

Paths & FFLAGS

configure.gsi

```
COREDIR       =  /d1/stark/GSI/src/pgi/release_V3.1
INC_DIR       =  $(COREDIR)/include
BYTE_ORDER    =  LITTLE_ENDIAN
SFC           =  pgf90
SF90          =  pgf90 -Mfree
SCC           =  gcc
INC_FLAGS     =  -module $(INC_DIR) -I $(INC_DIR) -I /
usr/local/netcdf3-pgi/include
```

Paths & FFLAGS

configure.gsi

```
COREDIR       =  /d1/stark/GSI/src/pgi/release_V3.1
INC_DIR       =  $(COREDIR)/include
BYTE_ORDER    =  LITTLE_ENDIAN
SFC           =  pgf90
SF90          =  pgf90 -Mfree
SCC           =  gcc
INC_FLAGS     =  -module $(INC_DIR) -I $(INC_DIR) -I /
usr/local/netcdf3-pgi/include
```

Paths & FFLAGS

configure.gsi

```
FFLAGS_DEFAULT =  -Kieee -pc 64 -Ktrap=fp
FFLAGS_DEBUG   =  -O0 -q -C
FFLAGS_OPT     =  -O3
FFLAGS         =  -fast $(FFLAGS_DEFAULT)
-DLANGUAGE_FORTRAN -DsysLinux
$(INC_FLAGS) $(LDFLAGS) -DLINUX –DPGI
```

configure.gsi

```
#### Architecture specific settings ####
# Settings for Linux x86_64, PGI compilers (pgf90 & gcc)  (dmpar,optimize)#
LDFLAGS       =  -Wl,-noinhibit-exec

COREDIR       =  /d1/stark/GSI/src/pgi/release_V3.1
INC_DIR       =  $(COREDIR)/include
BYTE_ORDER    =  LITTLE_ENDIAN
SFC           =  pgf90
SF90          =  pgf90 -Mfree
SCC           =  gcc
INC_FLAGS     =  -module $(INC_DIR) -I $(INC_DIR) -I /usr/local/netcdf3-pgi/include
FFLAGS_i4r4   =  -i4 -r4
FFLAGS_i4r8   =  -i4 -r8
FFLAGS_i8r8   =  -i8 -r8
FFLAGS_DEFAULT =  -Kieee -pc 64 -Ktrap=fp
FFLAGS_DEBUG  =  -O0 -q -C
FFLAGS_OPT    =  -O3
FFLAGS        =  -fast $(FFLAGS_DEFAULT) -DLANGUAGE_FORTRAN -DsysLinux $(INC_FLAGS) $(LDFLAGS) -DLINUX -DPGI
# Library build flags
FFLAGS_BACIO  =  -O3 $(FFLAGS_DEFAULT)
ARFLAGS_BACIO =
FFLAGS_BUFR   =  -O3 $(FFLAGS_DEFAULT) $(FFLAGS_i4r8)
CFLAGS_BUFR   =  -O3 -DUNDERSCORE
FFLAGS_CLOUD  =  -O3 $(FFLAGS_DEFAULT)
FFLAGS_CRTM   =  -fast $(FFLAGS_DEFAULT)
LFLAGS_CRTM   =
FFLAGS_GFSIO  =  -O3 $(FFLAGS_DEFAULT) $(FFLAGS_i4r4)
ARFLAGS_GFSIO =
```

Library Build FFLAGS

configure.gsi

```
# Library build flags
FFLAGS_BACIO   =  -O3 $(FFLAGS_DEFAULT)
ARFLAGS_BACIO  =
FFLAGS_BUFR    =  -O3 $(FFLAGS_DEFAULT)
$(FFLAGS_i4r8)
CFLAGS_BUFR    =  -O3 -DUNDERSCORE
FFLAGS_CLOUD   =  -O3 $(FFLAGS_DEFAULT)
FFLAGS_CRTM    =  -fast $(FFLAGS_DEFAULT)
LFLAGS_CRTM    =
FFLAGS_GFSIO   =  -O3 $(FFLAGS_DEFAULT)
$(FFLAGS_i4r4)
```

configure.gsi

```
CPP             =  cpp
CPP_FLAGS       =  -C -P -D$(BYTE_ORDER) -D_REAL8_ -
DWRF -DLINUX -DPGI
CPP_F90FLAGS    =  -traditional-cpp -lang-fortran

DM_FC           =  mpif90 -f90=pgf90
DM_F90          =  mpif90 -Mfree -f90=pgf90
DM_CC           =  gcc

CFLAGS          =  -O0 -DLINUX -DUNDERSCORE
CFLAGS2         =  -DLINUX -Dfunder -DFortranByte=char
-DFortranInt=int -DFortranLlong='long long'
```

Library Paths

configure.gsi

Main issues are:

- library paths are wrong
- library names are wrong

MYLIBsys = -llapack -lblas

NETCDF_PATH = -rpath,/usr/local/netcdf3-pgi/lib

Support

- For more detailed information on installation, please see:
  - GSI User's Guide, Chapter 2
    - www.dtcenter.org/com-GSI/users/docs/index.php
- For further assistance contact:
  - gsi_help@ucar.edu