Compiler diagnostic flags for Cheyenne users

Portability and correctness both are important goals when developing code. Non-standard code may not be portable, and its execution may be unpredictable.

Using diagnostic options when you compile your code can help you find potential problems. Since the compiler is going to analyze your code anyway, it pays to take advantage of the diagnostic options to learn as much as you can from the analysis.

Because of differences in compilers, it also is good practice to compile your code with each compiler that is available on the system, note any diagnostic messages you get, and revise your code accordingly.

The following options can be helpful as you compile code to run in the HPC environment that CISL manages.

Page contents

Intel

- **-debug all** – provides complete debugging information
- **-g** – places symbolic debugging information in the executable program
- **-check all** – performs all runtime checks (includes bounds checking)
- **-warn all** – enables all warnings
- **-stand f08** – warns of usage that does not conform to the Fortran 2008 standard
- **-traceback** – enables stack trace if the program crashes

Also see Intel C++ diagnostic options.

GNU

- **-ggdb** – places symbolic debugging information in the executable program for use by GDB
- **-fcheck=all** – performs all runtime checks (includes bounds checking)
- **-Wall** – enables all warnings
- **-std=f2008** – warns of usage that does not conform to the Fortran 2008 standard

Also see GCC diagnostic warning options.

NVIDIA HPC (PGI)

- **-C** – performs runtime bounds checking
- **-g** – places symbolic debugging information in the executable program
- **-traceback** – enables stack trace if the program crashes