

WIKI for Hyak users-Building Complex Codes

Specific Issues Related to Building Complex Codes

We provide all Hyak users with a complete shell and software development environment comprising:

- Bash shell environment, including `.bashrc`, `.bash_profile` and `.bash_logout` files. If you ever want to start fresh, you can copy a new set over from `/etc/skel` on a login node.
- [Modules](#) for a variety of software development tools and applications. Modules set paths and various environment variables required to use specific software.
- Complete [Intel](#) and GNU development environments. These include C and Fortran compilers, math and MPI libraries, debuggers, and optimization tools.
- Detailed [examples](#) in the configuration and use of every element of the development environment.

For software packages using GNU autoconf to manage the build process, it is often possible to set up your environment using the modules we provide and have everything else work automagically. Unfortunately, much scientific and engineering software fails to follow anything like standard software engineering practices, and things get complicated. When you encounter a package which uses unconventional makefiles, hard-coded paths and variables in non-standard locations, etc., we suggest you:

- Configure your environment as usual, using the modules we provide. For example, load the module for the compiler and MPI library (i.e. `icc_X.X-impi_X.X`) of your choice.
- Display the content of the module using `module show <module>` and use these variables to set up your makefiles or other build setup files.
- Use bash as your shell. You may, of course, use the C shell if you like. However, the qsub examples we provide are all bash based, and some users have reported issues with their C shell environments not transferring completely to the bash environment when they submit scripts based on our bash examples to the scheduler.