

Ranger User Guide

Environment Section

Login Shell

The most important component of a user's environment is the login shell that interprets text on each interactive command line and statements in shell scripts. Each login has a line entry in the `/etc/passwd` file, and the last field contains the shell launched at login. To determine your login shell, execute:

```
echo $SHELL {to see your login shell}
```

You can use the `chsh` command to change your login shell; instructions are in the man page. Available shells are listed in the `/etc/shells` file with their full-path. To change your login shell, execute:

```
cat /etc/shells           {select a <shell> from list}
chsh -s <shell> <username> {use full path of the shell}
```

User Environment

The next most important component of a user's environment is the set of environment variables. Many of the Unix commands and tools, such as the compilers, debuggers, profilers, editors, and just about all applications that have GUIs (Graphical User Interfaces), look in the environment for variables that specify information they may need to access. To see the variables in your environment execute the command:

```
env {to see environment variables}
```

The variables are listed as keyword/value pairs separated by an equal (=) sign, as illustrated below by the HOME and PATH variables.

```
HOME=/home/utexas/staff/milfeld
PATH=/bin:/usr/bin:/usr/local/apps:/opt/intel/bin
```

(PATH has a colon (:)) separated list of paths for its value.) It is important to realize that variables set in the environment (with `setenv` for C shells and `export` for Bourne shells) are "carried" to the environment of shell scripts and new shell invocations, while normal "shell" variables (created with the `set` command) are useful only in the present shell. Only environment variables are seen in the `env` (or `printenv`) command; execute `set` to see the (normal) shell variables.

Startup Scripts

All Unix systems set up a default environment and provide administrators and users with the ability to execute additional Unix commands to alter the environment. These commands are "sourced"; that is, they are executed by your login shell, and the variables (both normal and environmental) as well as aliases and functions are included in the present environment. We recommend that you customize the login environment by inserting your "startup" commands in `.cshrc_user`, `.login_user`, and `.profile_user` files in your home directory.

Basic site environment variables and aliases are set in

```
/usr/local/etc/cshrc {C-shell, non-login specific}
```

```
/usr/local/etc/login {C-shell, specific to login}
```

```
/usr/local/etc/profile {Bourne-type shells}
```

For historical reasons, the C shells source two types of files. The **.cshrc** type files are sourced first (**/etc/csh.cshrc--> \$HOME/.cshrc--> /usr/local/etc/cshrc--> \$HOME/.cshrc_user**). These files are used to set up environments that are to be executed by all scripts and used for access to the machine without a login. For example, the following commands only execute the **.cshrc** type files on the remote machine:

```
scp data ranger.tacc.utexas.edu: {only .cshrc sourced on ranger}
```

```
ssh ranger.tacc.utexas.edu date {only .cshrc sourced on ranger}
```

The **.login** type files are used to setup environment variables that you commonly use in an interactive session. They are sourced after the **.cshrc** type files (**/etc/csh.login--> \$HOME/.login--> /usr/local/etc/login--> \$HOME/.login_user**). Similarly, if your login shell is a Bourne shell (bash, sh, ksh, ...), the profile files are sourced (**/etc/profile--> \$HOME/.profile--> /usr/local/etc/profile--> \$HOME/.profile_user**).

The commands in the **/etc** files above are concerned with operating system behavior and set the initial **PATH**, **ulimit**, **umask**, and environment variables such as the **HOSTNAME**. They also source command scripts in **/etc/profile.d** -- the **/etc/csh.cshrc** sources files ending in **.csh**, and **/etc/profile** sources files ending in **.sh**. Many site administrators use these scripts to setup the environments for common user tools (vim, less, etc.) and system utilities (ganglia, modules, Globus, LSF, etc.)

TACC has to coordinate the environments on platforms of several operating systems: AIX, Linux, IRIX, Solaris, and Unicos. In order to efficiently maintain and create a common environment among these systems, TACC uses its own startup files in **/usr/local/etc**. (A corresponding file in this **etc** directory is sourced by the **.profile**, **.**, and **.login** files that reside in your home directory. (Please do not remove these files and the sourcing commands in them, even if you are a Unix guru.) Any commands that you put in your **.login_user**, **.cshrc_user**, or **.profile_user** file are sourced (if the file exists) at the end of the corresponding **/usr/local/etc** command files. If you accidentally remove your **.login**, **.cshrc**, and **.login**, you can copy new ones from **/usr/local/etc/start-up** or execute

```
/usr/local/bin/install_ut_startups
```

to get a new copy (your old files are renamed with a date suffix).

Modules

TACC is constantly including updates and installing revisions for application packages, compilers, communications libraries, and tools and math libraries. To facilitate the task of updating and to provide a uniform mechanism for accessing different revisions of software, TACC uses the modules utility.

At login, a basic environment for the default compilers, tools, and libraries is set by several modules commands. Your **PATH**, **MANPATH**, **LIBPATH**, directory locations (**WORK**, **HOME**, ...), aliases (**cdw** ...) and license paths, are just a few of the environment variables and aliases created for you. This frees you from having to initially set them and update them whenever modifications and updates are made in system and application software.

Users who need 3rd party applications, special libraries, and tools for their development can quickly tailor their environment with only the applications and tools they need. (Building your own specific application environment through modules allows you to keep your environment free from the clutter of all the other application environments you don't need.)

Each of the major TACC applications has a **modulefile** that sets, unsets, appends to, or prepends to environment variables such as **\$PATH**, **\$LD_LIBRARY_PATH**, **\$INCLUDE_PATH**, **\$MANPATH** for the specific application. Each **modulefile** also sets functions or aliases for use with the application. A user need only invoke a single command,

```
module load <application>
```

at each login to configure an application/programming environment properly. If you often need an application environment, place the modules command in your **.login_user** and/or **.profile_user** shell startup file.

Most of the package directories are in **/opt/apps (\$APPS)** and are named after the package name (**<app>**). In each package directory there are subdirectories that contain the specific versions of the package.

As an example, the **fftw** package requires several environment variables that point to its home, libraries, include files, and documentation. These can be set in your environment by loading the **fftw** module:

module load fftw

To see a list of available modules and a synopsis of a **modulefile**'s operations, execute:

```
module available {lists modules}
module help <app> {lists environment changes performed for <app>}
```

During upgrades, new **modulefiles** are created to reflect the changes made to the environment variables. TACC will always announce upgrades and module changes in advance.

File Systems

The TACC HPC platforms have several different file systems with distinct storage characteristics. There are predefined, user-owned directories in these file systems for users to store their data. Of course, these file systems are shared with other users, so they are managed by either a quota limit, a purge policy (time-residency) limit, or a migration policy.

Three Lustre file systems are available to users: **\$HOME**, **\$WORK** and **\$SCRATCH**. Users have 6GB for **\$HOME**. **\$WORK** on our Ranger system is NOT a purged file system, but is limited by a large quota. Use **\$SCRATCH** for temporary, large file storage; this file system is purged periodically (TBD), and has a very large quota. All file systems also impose an inode limit.

To determine the size of a file system, **cd** to the directory of interest and execute the "**df -k .**" command.

Without the "**dot**" all file systems are reported. In the **df** command output below, the file system name appears on the left (IP number, "ib" protocol, using OFED gen2), and the used and available space (**-k**, in units of 1KBytes) appear in the middle columns followed by the percent used and the mount point:

```
% df -k .
```

File System	1k-blocks	Used	Available	Use%	Mounted on
129.114.97.1@o2ib:/share	103836108384	290290196	103545814376	1%	/share

To determine the amount of space occupied in a user-owned directory, **cd** to the directory and execute the **du** command with the **-sb** option (s=summary, b=units in bytes):

```
du -sb
```

To determine quota limits and usage, execute the **lfs quota** command with your username and the directory of interest:

```
lfs quota -u <username> $HOME
```

```
lfs quota -u <username> $WORK
```

```
lfs quota -u <username> $SCRATCH
```

The four major file systems available on Ranger are:

home directory

At login, the system automatically changes to your home directory.

This is the recommended location to store your source codes and build your executables.

The **quota limit** on home is **6GB**.

A user's home directory is accessible from the frontend node and any compute node.

Use **\$HOME** to reference your home directory in scripts.

Use **cd** to change to **\$HOME**.

work directory

Store large files here.

Often users change to this directory in their batch scripts and run their jobs in this file system.

A user's work directory is accessible from the frontend node and any compute node.

The user quota is **350GB**.

Purge Policy: There is no purging on this file system.

This file system is not backed up.

Use **\$WORK** to reference your work directory in scripts.

Use **cdw** to change to **\$WORK**.

-->

scratch or temporary directory

This file system is not on line yet.

UNLIKE the TACC Lonestar system, this is **NOT** a local disk file system on each node.

This is a global Lustre file system for storing temporary files.

Purge Policy: Files on this system are purged when a file's access time exceeds **10** days.

PLEASE NOTE: TACC staff may delete files from scratch if the scratch file system becomes full and directories consume an inordinately large amount of disk space, even if files are less than **10** days old. A full scratch file system inhibits use of the file system for all users. The use of programs or scripts to actively circumvent the file purge policy will not be tolerated.

Often, in batch jobs it is more efficient to use and store files directly in **\$WORK** (to avoid moving files from **scratch** later before they are purged).

The quota on this system is 400TB.

Use **\$SCRATCH** to reference this file system in scripts.

Use **cds** to change to **\$SCRATCH**.

archive

[More on Archive -- rls and sync not available yet](#)