

Modules

Introduction

To simplify setting up environments for compilers, MPI-versions and applications, Colossus has the Environment Modules package installed. This provides a means for dynamic modification of a user's environment via *modulefiles*.

Each modulefile contains the information needed to configure the shell for an environment. Typically, a modulefile modifies environment variables like PATH, MANPATH, LD_LIBRARY_PATH, CPATH and PKG_CONFIG_PATH to give access to an application or library. To invoke a module file you run the `module` command with arguments.

In TSD, the `module` command can be used from the command line on submit hosts and as part of a job script when submitting a job to Colossus. You can reach the submit host assigned to your project by connecting to `pXX-submit.tsd.usit.no` using PUTTY from your Windows VM, or by using `ssh` from your linux VM.

Listing available modules

The available modulefiles can be listed by `module avail`. Example output:

```
$ module avail
----- /cluster/software/MODULEFILES/custom -----
atlas-lapack/3.10.3      matlab/R2018b
bcftools/1.8            netcdf-fortran.gnu/4.4.4
bedtools/2.17.0        netcdf.gnu/4.6.1
bismark/0.22.1         openmpi.gnu/1.8
bowtie/1.2.2           openmpi.gnu/3.1.3
[...]
----- /cluster/software/MODULEFILES/easybuild/all -----
AFNI/18.3.00-foss-2018b-Python-3.6.6  libreadline/7.0-GCCcore-6.4.0
Anaconda3/5.3.0                       libreadline/7.0-GCCcore-7.3.0
arpack-ng/3.7.0-foss-2019a            libreadline/8.0-GCCcore-8.2.0
[...]
```

You'll see *custom* and *Easybuild* modules. The Easybuild modules are usually the most recent and should be preferred. When loading a custom module a notification will be displayed to clarify this.

You can also specify a name:

```
$ module avail MATLAB
----- /cluster/software/MODULEFILES/easybuild/all -----
  MATLAB/2017b    MATLAB/2018a    MATLAB/2018b    MATLAB/2019a
  MATLAB/2019b    MATLAB/2020a    MATLAB/2020b
```

Loading modules

Modulefiles can be used from most shells, as well as Perl and Python. Modulefiles can be loaded or unloaded on the command line or within your scripts by typing:

```
module load software/version
module unload software/version
```

If a version is not specified, no software will be loaded. For the purpose of reproducibility, we enforce users to load specific versions of modulefiles.

You will get a list of currently loaded modulefiles by:

```
$ module list
Currently Loaded Modulefiles:
 1) MATLAB/2020b
```

To view what exactly a modulefile is doing to your environment, type:

```
module show modulefile/version
```

Switch to a different version of a module

Switching to another version is similar to loading a specific version. As an example, if you want to switch from the current loaded R-bundle-Bioconductor to an more recent one:

```
module switch R-bundle-Bioconductor/3.9-foss-2019a-R-3.6.0 R-bundle-Bioconductor/3.9-foss-2019a-R-3.6.0
```

This, more compact syntax will fortunately also work:

```
module switch R-bundle-Bioconductor R-bundle-Bioconductor/3.11-foss-2020a-R-4.0
```

Loading multiple modules simultaneously

A module loads all necessary dependencies along with the parent module. This can lead to conflicts when different modules have conflicting dependencies, e.g. they depend on conflicting versions of the tool chains (`foss`, `GCC`, `GCCcore`). We recommend loading the modules sequentially and resolve any conflicts. When modules get reloaded with a version change this indicates a conflict and will break the dependencies of the previously loaded module(s), e.g.:

```
module load R-bundle-Bioconductor/3.11-foss-2020a-R-4.0.0
module load libGLU/9.0.0-foss-2018b
```

The following have been reloaded with a version change:

- 1) FFTW/3.3.8-gompi-2020a => FFTW/3.3.8-gompi-2018b
- 2) GCC/9.3.0 => GCC/7.3.0-2.30
- 3) GCCcore/9.3.0 => GCCcore/7.3.0
- 4) GMP/6.2.0-GCCcore-9.3.0 => GMP/6.1.2-GCCcore-7.3.0

Purge modules

We recommend using

```
module purge
```

prior to any `module load` commands in job scripts, to prevent inheriting environment variables set by `module` commands in the shell used when submitting the job.

Help

A help text exists for most of the modulefiles. To view it, type:

```
module help modulefile/version
```

For further documentation, please refer to the [Environment Modules homepage](#).

Personalize default modulefiles

You may add loading of modulefiles to your `.bash_login` file to make sure your favorite modulefile is always loaded when logging in. For instance, to always have the MATLAB/2020b modulefile load on login, add:

```
module load MATLAB/2020b
```

to the `.bash_login` file.



Contact support



Chat



Register
case



Video call



Phone

Did you find what you were looking for?

[Give us feedback](#)

Published June 21, 2021 10:35 AM - Last modified Feb. 24, 2022 2:41 PM



Share by
E-mail



Share on
Facebook



Share on
Twitter



UNIVERSITY
OF OSLO

CONTACT
INFORMATION

[Contact us](#)
[Find us](#)

ABOUT THE
WEBSITE

[Cookies](#)
[Accessibility
statement](#)
[\(in
Norwegian
only\)](#)

RESPONSIBLE FOR
THIS PAGE

[Web editor USIT](#)
[Log in](#) | 