

Environment modules TCL



DEPRECATED

Due to update of the computing nodes operating system this documentation page will be obsolete. The new tool will be [Lmod](#)

You are in an environment shared by hundreds of users, so the way to provide multiple applications in their multiple versions to a multitude of users in a distributed computing environment is not the same as on a personal computer or your own server.

The tool for managing these environments is called ***Environment Modules***. You can find an extensive explanation about Environment Modules at the following links:

- <http://modules.sourceforge.net>
 - <https://modules.readthedocs.io/en/stable/module.html>

Basically **Environment modules** are a **set of scripts** that allow us to easily modify the user environment (the shell) making available to the user only the applications we want to use. For these reason, it will be the tool to **load / unload applications** you want to use.

How to use environment modules?

You will see in later sections how they are used in conjunction with job scheduler **Slurm**. To familiarize yourself with this tool you can [log in](#) and run the following commands and examples to understand how it works:

`$module avail (o av)` -- shows available modules

\$module load modulename (o add) -- load a specific module / application

\$module unload modulename (or rm) -- remove a loaded module

`$module purge` -- remove all loaded modules

`$module list` -- list the modules loaded by the user

```
$module show modulename          -- show module information  
  
$module whatis modulename        -- show modulefile information  
  
$module update modulename        -- reload module  
  
$module help                      -- show module help
```

Examples

```
$ which gcc           -> /usr/bin/gcc  
$ module load gcc/10.2.0  
$ which gcc           -> /opt/envhpc/utils/rhel6/gcc/10.2.0/bin/gcc
```

You can see how the application executable is being used before and after loading the module.

```
$ module av python  
  
----- /opt/envhpc/modulefiles/.rhel6 -----  
python/2.7.18/gcc python/3.5.4/gcc python/3.7.9/gcc python/3.8.11/gcc  
  
$ which python           -> /usr/bin/python  
$ which python3          -> /usr/bin/which: no python3 in (/usr/local/bin,...)  
  
$ module load python/3.7.9/gcc -> Requisito cargar openssl/1.1.1k/gcc  
  
python/3.7.9/gcc(10):ERROR:151: Module 'python/3.7.9/gcc' depends on one of the  
module(s) 'openssl/1.1.1k/gcc'.  
python/3.7.9/gcc(10):ERROR:102: Tcl command execution failed: prereq openssl/1.1.1k/gcc.  
  
$ module load openssl/1.1.1k/gcc python/3.7.9/gcc  
$ which python3          -> /opt/envhpc/utils/rhel6/python/3.7.9/gcc-10.2.0/bin/python3
```

As first step, look for some version of python available on the system.

Next, you will notice that *python3* command is not available.

When loading the python module, say *openssl / 1.1.1k / gcc* is a requirement to load *python/3.7.9/gcc*.

After these two modules are loaded, you can see how python version 3.7.9 is available.

[How to use modules modules with a script?](#)