Modules for Experiments in Stellar Astrophysics (MESA)

A note up front: this is not the only piece of software called “MESA”, and also not the most common one.

With that out of the way, the MESA package is used in several courses and research projects.

Pre-installed versions

On our systems, we have a couple of versions of MESA and the MESASDK already installed. So if that suits your needs, you can get started by loading the module. Actually, loading the `mesa` module alone will be sufficient, since this will automatically load the `mesasdk` as a dependency. E.g:

```
module load mesa/r24.08.1
```

or to load the current default version (usually the latest version):

```
module load mesa
```

Own setup

You can follow the instructions as found on https://docs.mesastar.org/en/latest/installation.html#installing-mesa but DON'T include the `source $MESASDK_ROOT/something` command in your session startup files (see below for the reason, and better alternatives). However, setting environment variables in your session files is safe, and can be very useful so you don't have to remember where your personal install is located.

A mix

It is possible to use a pre-installed version of MESASDK, but maintain your own copy of MESA. In that case, load one of the available mesasdk modules, but point environment variable MESA_DIR to your own copy of MESA.

No setup needed, nor wanted!!

Whatever method you choose for using MESA, please don't follow instructions found elsewhere (including the official documentation, and older course notes) to run the MESA setup in your session startup files like `.bashrc` or `.tcshrc`.

The reason for this is that MESASDK is a complete environment with compilers, libraries and various tools, exactly suited to build and work with a specific version of MESA. But these tools are different versions than the ones needed by anything else on the system. So with MESASDK loaded into your environment, you may not be able to compile other code, but it has been known even to interfere with pre-installed software that requires different versions of some of these tools or libraries.

Note that the official instructions state exactly the same thing, warning you against including these commands, unless you know what you are doing (the instructions even point to better alternatives)!!

So please DON'T include the 'module load' or 'source' commands in a startup file, just type them when needed.

To make life easier, you can define a command alias so you can activate the MESA environment with a single command, e.g. in `.bashrc`:

```
alias mesainit="module load mesa/r24.08.1"
```

or in `.tcshrc`:

```
alias mesainit "module load mesa/r24.08.1"
```

For your personal MESA installs, the same method can be used, with the alias pointing to the `source` command needed to initialize the setup. As mentioned earlier, setting environment variables is safe to do, so that can be done here as well: For users of the bash shell:

```
export MESASDK_ROOT=/net/vdesk/data2/your_username/MESA/mesasdk
export MESA_DIR=/net/vdesk/data2/your_username/MESA/mesa-r10398
alias mesainit "source $MESASDK_ROOT/bin/mesasdk_init.sh"
```

or for the tcsh shell:

```
setenv MESASDK_ROOT /net/vdesk/data2/your_username/MESA/mesasdk
setenv MESA_DIR /net/vdesk/data2/your_username/MESA/mesa-r10398
alias mesainit "source $MESASDK_ROOT/bin/mesasdk_init.csh"
```

Of course, make sure you replace the path names from this example with the actual location where you have installed the software!

And, when editing session startup files like `.bashrc` or `.tcshrc`, it is always a good idea to test them in your current session, before logging out to have them in effect for your next login. The reason is, that mistakes are easy to make, and many login environments are not very forgiving, ending the login attempt with an error if there is an error in the startup files. A simple way to test your setup is to run the shell from the command line with the `-l` option to process login files:

```
''tcsh -l''
```

or

```
''bash -l''
```

Any errors will be displayed in your terminal, so you can fix them before they block you from further logins.

Yet another methoud would be, to include the setenv (or export) commands and the source command into a small text file or script, and load that file when you want to activate the MESA environment, e.g create file `mymesa` containing:

```
setenv MESASDK_ROOT /net/vdesk/data2/your_username/MESA/mesasdk
setenv MESA_DIR /net/vdesk/data2/your_username/MESA/mesa-r10398
source $MESASDK_ROOT/bin/mesasdk_init.csh
```

And when you want to active your MESA environment, type `source mymesa`. Like the method with the `alias`, this method makes sure the environment is only loaded when needed, so it will not interfere with normal logins or with other programs.

Links

See also:

- MESA website
- MESA on github

From:
https://helpdesk.strw.leidenuniv.nl/wiki/ - **Computer Documentation Wiki**

Permanent link:
https://helpdesk.strw.leidenuniv.nl/wiki/doku.php?id=mesa

Last update: **2025/04/30 11:37**