

The Lofar Login Environment (LLE)

This page describes the standard login environment for new users on the LOFAR cluster nodes. Existing accounts can be easily modified; see below for a manual to do this.

General

The Lofar Login Environment sets a few things for your account, and allows you to easily initialise often used packages and tools at login time. To this purpose we provide a few simple standardized login-scripts.

Use (t)csH or bash!

To be able to use the LLE environment you must either use a (t)csH or bash login shell. We do not support scripts for any other shells.

What to do when you get an account

(t)csH

If you have chosen for the (t)csH shell, you do as follows:

- Log in and go to your (empty) \$HOME directory
- `> ln -s /opt/login/cshrc .cshrc` (Note: on lioffen you should refer to /app/login/cshrc !)
- Log out and login again; you should see a welcome message.

bash

If you have chosen for the bash shell, you do as follows:

- Log in and go to your (empty) \$HOME directory
- `> ln -s /opt/login/bashrc .bashrc` (Note: on lioffen you should refer to /app/login/cshrc !)
- Log out and login again; you should see a welcome message.

How to transform existing accounts?

To use the LLE in an existing account, rename your existing .cshrc or .bashrc in your \$HOME and follow the instructions given above.

Make sure that possibly existing files `$HOME/.login`, `$HOME/.setenv` and `$HOME/.alias` that you still want to use at login time are renamed to `$HOME/.mylogin`, `$HOME/.mysetenv`, `$HOME/.myalias`.

The LLE scripts

Some of the details are presented here.

What is `$APS_LOCAL` ?

The root directory for the scripts is provided in environment variable `$APS_LOCAL`, whose value is set in the `.cshrc` and `.bashrc` scripts. The actual value depends on your location:

- New CEP processing cluster: `/opt`
- lioffen and its nodes: `/app`

Login scripts

In directory `${APS_LOCAL}/login` are a number of default login scripts. The scripts ending with `.bash` are for the bash shell, the others for the (t) csh shell:

- `cshrc` → sets `APS_LOCAL` and calls the other scripts
- `login` → Displays welcome message; no settings
- `setenv` → adds some items to `$PATH`, sets your prompt, etc.
- `alias` → some default aliases to make life easy
- `setpackages` → script to define packages to initialize at login-time

Users should **NEVER** modify these default scripts at this location!

Personal command aliases: `.myalias`

Personal command aliases can be added to a file `$HOME/.myalias`. When this file exists, the `.cshrc` or `.bashrc` script will read this file after reading the default `${APS_LOCAL}/login/alias(.bash)`.

Personal environment settings: `.mysetenv`

Personal extensions to `$PATH`, personal environment variables, personal prompt setting, or overloaded existing environment variables should be done in a file `$HOME/.mysetenv`. If this file exists, the `cshrc` or `bashrc` script will read this file.

Package initialisation: .mypackages

To initialize the use of installed packages on your system you must create a file `$HOME/.mypackages`. If this file exists, it is used by the LLE script `setpackages`.

In the directory `${APS_LOCAL}/scripts` you will find available package initialization scripts, named like `do<package>` (e.g., `doCasacore`, `doPyrap`). Whenever you add a package to the file `$HOME/.mypackages`, the associated script in `$APS_LOCAL/scripts` will be source'd.

The file `$HOME/.mypackages` can look like this example:

```
Casacore
doPyrap
```

The packages can be on one line or on multiple lines. Also note the use of `<package>` and `do<package>`; both are accepted. If an initialization script cannot be found, it will be reported to the user.

How to add a personal package

Apart from the systemwide `do<package>` files provided in directory `$APS_LOCAL/scripts`, users can add their personal `do<package>` files in their `$HOME` and have these run at login time. If there is a `do<package>` file both in `$HOME` and in `$APS_LOCAL/scripts`, the version in `$HOME` has preference and will be executed. This can be used to test a different version of a package, etc... To add a new, personal, package, act as follows:

- Install the package
- Create a `$HOME/do<package>` initialization script (e.g., adding the installation directory to your `$PATH`)
- Add `<package>` to the list of packages in file `$HOME/.mypackages`

Starting X environment

To start up an X environment you have to make sure that

- The files `$HOME/.xinitrc` and `$HOME/.xsession` are removed

Determine the colordepth that you need (8, 16, 24 bit display) and type:

```
startx -- :1 -depth <colordepth>
```

This should start your X environment at the proper colordepth. This will create an additional X Server next to the one you are already running. You can access this with `Alt-F8` or `Cntl-Alt-F8`. The `F7` variant will give you back your normal X.

Help!

If you need help, or have questions, or want to give any other comments, contact Arno Schoenmakers or Adriaan Renting.

New do<xxxx>-files can be added at any time if you have a new package that should be available systemwide. Contact Arno Schoenmakers or Adriaan Renting in this case.

From:

<https://www.astron.nl/lofarwiki/> - **LOFAR Wiki**

Permanent link:

<https://www.astron.nl/lofarwiki/doku.php?id=public:lle&rev=1254236833>

Last update: **2009-09-29 15:07**

