LOFAR User Software

TOC:

Code repository

Organization of the repository

```
usg.lofar.org/svn
|-- code
|   |-- branches
|   `-- trunk
|   |   |-- devel_common
|   |   |-- data
|   |   |-- external
|   |   `-- src
`-- documents
    |-- branches
    `-- trunk
```

Checking out code

As read-only access to the repository is not restricted in any ways, you can obtain a working copy of the source code by running

- using Subversion...

```
svn co http://usg.lofar.org/svn/code/trunk lofarsoft
```

- using Git...

```
git svn clone http://usg.lofar.org/svn/code/trunk lofarsoft
```

Please be aware though that this will retrieve to complete backlog of all changes, so you might rather use

```
git svn clone -r <revision> http://usg.lofar.org/svn/code/trunk
lofarsoft
cd lofarsoft
git svn rebase
```

where <revision> is either a specific revision number or the word HEAD, which refers to the latest available version.

In case you not only want a working version of the source code, but also of the various documents, you do have two options to options of retrieval:

1. Check out everything in a single go:

```
svn co http://usg.lofar.org/svn usg
```

2. Check out a slightly cleaned-up version, omitting the trunk directories from your working version:

```
mkdir usg
cd usg
svn co http://usg.lofar.org/svn/code/trunk code
svn co http://usg.lofar.org/svn/documents/trunk docs
```

Bootstrapping your working copy

Once the checkout from the central repository has completed, you are left with a directory structure as described above. The next step now is to get to the point where it is possible to build (and subsequently install) packages in the software collection…

As the LUS uses the CMake cross-platform makefile generator to handle the configuration of the code base, the most important thing to check is whether or not a suitable version of CMake is available on your platform – this check is carried out through the bootstrap script in the top-level directory of the working copy:

```
./bootstrap
```

The the main job of the bootstrap script is to check whether or not a recent enough version of CMake is installed on your system; if the found installation is too old or not available at all, a build from the provided sources is triggered and the resulting executables will be installed into “release/bin” (which of course then should be in your PATH).

Updating your working copy

As with the case of retrieving your working working copy from the central repository, this step is very much depending on the tool used locally for version control:

- Subversion : Change back to the top-level directory of your working copy and and run

```
svn up
```

- Git : Depending on your choice to deal with upstream changes, you might change back to the `master` branch of your working copy before running

```
git checkout master
git svn rebase
```

  If however you prefer to directly merge the upstream changes into your feature/development branch, you simply run

```
git svn rebase
```

  In either case ensure there are no uncommitted changes – either add and then commit them or stash them, before pulling in the upstream changes.

Write access to the repository

<html><font color=red>This information is outdated! It will be updated soon</html>

While (by design) the user software repository is world-wide readable, write access is being restricted to a list of registered users. The basic procedure for getting added to that list – which basically relies on a combination of a `username` and MD5 encrypted `password` – is described below:

The information which needs to be provided by the user is a combination of `username` and `password`, where the latter is being hashed using the MD5 algorithm. The encryption of the password can be done in a number of ways, depending on the tools available to the user requesting access:

- Using `htpasswd`:

```
htpasswd -nbm <username> <password>
```

- Using `openssl`:

```
openssl passwd -apr1 <password>
```

If none of the above mentioned tools are available, use can be made of an online htpasswd generator.

Depending on the command line tool being used, the output will contain the full string to be entered

into the password file or the encrypted password only (in which case the username needs to be prepended):

```
lbaehren:$apr1$ziNPu...$YYKeohAqIiIzfz4YA12345    ## htpasswd
$apr1$9H8IBSvy$yswI9jLosDkDx1a6.12345             ## openssl
```

Supported platforms

First-level supported platforms:

1. Ubuntu 10.04 LTS

Second-level supported platforms:

1. Debian GNU/Linux 6.x