

User Software

[Code repository](#) | [Software packages](#)

Code repository

1. [Organization of the repository](#)
2. [Checking out code](#) / Read access
3. [Updating your working copy](#)
4. [Write access to the repository](#)

Organization of the repository

```
lus.lofar.org/svn
|-- code
|   |-- branches
|   `-- trunk
|       |-- build
|       |-- 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

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

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://lus.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://lus.lofar.org/svn/code/trunk code
```

```
svn co http://lus.lofar.org/svn/documents/trunk docs
```

Updating your working copy

Go to the build directory and type

```
make update
```

In the simplest case this might be nothing but a wrapper around the update command of Subversion, but further actions might be carried out if necessary.

Then build your target, for example “dal” with

```
make dal
```

You can also just build the target folder leaving everything else untouched using

```
cd dal  
make rebuild_cache && make && make install
```

Write access to the repository

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:

Software packages

- The [Data Access Library](#) (DAL) is a library that abstracts the underlying file format from the user when working with radio astronomy data. Underlying formats may include HDF5, AIPS++/CASA tables, FITS or specific raw formats. This library is available to both the C/C++ developer and the Python user via the “pydal” module.
- [CR-Tools](#)
- [Pulsar Tools](#)
- [pyBDSM](#)

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

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=public:user_software:user_software&rev=1318930874

Last update: **2011-10-18 09:41**

