

User Software :: CR-Tools :: Installation

1. [Required external software](#)
2. [Obtaining the source code](#)
3. [Configure and build](#)
 - [a\) Quick install](#)
 - [b\) Manual install](#)
 - [c\) Configuration options](#)
 - [d\) Environment variables](#)

Required external software

The following components should be installed on your machine:

Package	Comment	Max OS X	Debian Linux
Subversion client		Fink	
C/C++ compiler			
Fortran compiler			
Flex		Fink	
Bison		Fink	
LAPACK		Fink	
BLAS		Fink	
SWIG	required for PLplot		
Python	application and development files		
gawk	v3.1.6 or higher		
bzip2	gave problems when missing on a debian system		install libbz2-dev

The following external components are required but come as part of the provided distribution:

- [CMake](#) - cross-platform make generator
- [WCSSLIB](#) - World Coordinate Systems handling for astronomical images
- [CFITSIO](#) - I/O with FITS data
- [casacore](#) - Core libraries of CASA
- HDF5 library
- Boost++ *Python* module
- Python modules:
 - numarray
 - matplotlib
- PLplot

Finally there are a number of (more or less) *optional components*, which depend on the presence of the following packages:

- Blitz++ - some of the vector conversion routines also support working with Blitz++ arrays; default settings however use CASA classes
- LOPES-Star - when trying to build the contents of this module, the following additional packages will be required:
 - ROOT

- FFTW3 (note the difference from FFTW!)
- (Experience showed that we don't always manage to make the dependence on LOPES-Star optional. So if you are only interested in a running version we recommend installing ROOT and FFTW3.)

Obtaining the source code

All the source code for the [CR-Tools](#) is available through the Subversion repository of the User Software Group; in order to obtain a working copy run the following command:

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

Configure and build

a) Quick install

There is a build script available, which will go through the list of required external components – and finally build the CR-Tools themselves. The script can be run in two different ways:

1. Set environment variables for the code tree:

```
cd lofarsoft
export LOFARSOFT=`pwd`
. devel_common/scripts/init.sh
```

2. Direct call to the script in the build directory:

```
cd $LOFARSOFT/build
./bootstrap
make cr
```

Notes:

- If this fails and/or you need more control over the configuration options, you should have a look at the optional control parameters at the end of the next section.
- If you want a clean start after a failed build – e.g. after debugging inside the configuration scripts – the safest way to do this is to remove the previously created package directories inside the build directory:

```
make clean_build
```

b) Manual install

```
cd usg/build
```

```
cmake ../src/CR-Tools
make install
```

Though of course some details may look different on your system, you should be receiving an output like this:

```
-- The C compiler identification is GNU
-- The CXX compiler identification is GNU
-- Check for working C compiler: /usr/bin/gcc
-- Check for working C compiler: /usr/bin/gcc -- works
-- Check size of void*
-- Check size of void* - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- [CR] Configuration to use CASA ...
-- [CR] Installation area located.
-- CMAKE_SYSTEM ..... : Linux-2.6.4-52-smp
-- CMAKE_SYSTEM_PROCESSOR : i686
-- CMAKE_SYSTEM_64BIT ..... : 0
-- CMAKE_SYSTEM_BIG_ENDIAN :
-- BLITZ ..... : TRUE
-- CASA ..... : TRUE
-- CASACORE ..... : FALSE
-- CFITSIO ..... : TRUE
-- FFTW3 ..... : TRUE
-- LAPACK ..... : TRUE
-- PGPLOT ..... : TRUE
-- ROOT ..... : TRUE
-- WCSLIB ..... : TRUE

-- Configuring done
-- Generating done
-- Build files have been written to: /opt/local/lofarsoft/build/cr
```

Depending on what is already installed on your system, you will get a number of warnings or error messages stating that required external packages are missing; a common scenario e.g. is, that you are missing CFITSIO.

c) Configuration options

The list of configuration options (e.g. build the tools collection **with** Glish support) can be found at the [optional build parameters](#) page.

(Glish support is needed to build the Glish clients needed for the “old” lopestools. It requires a working CASA installation that was installed separately.)

d) Environment variables

Add the location of the new executables to your PATH variable:

- For *csh* and *tcsh*:

```
set path = ( $path <LOFARSOFT>/release/bin )
```

In this LOFARSOFT should point to base of your working copy of the USG code tree.

- For *sh* and *bash* add the following lines to your *.profile* or *.bashrc* file:

```
export LOFARSOFT=<path to root working copy>
. $LOFARSOFT/devel_common/scripts/init.sh
```

Even though it is not expected that the init instruction will be undergoing a lot of changes, this is the securest method by which to catch changes if they take place.

If you don't have matplotlib or scipy installed you can install them from the repository:

```
cd $LOFARSOFT/build
make matplotlib
make scipy
```

← [User Software](#) • [CR-Tools](#)

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

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=public:user_software:cr-tools:installation_of_cr-tools

Last update: **2017-03-08 15:27**

