Lofar Software Stack

This page gives an overview of the Lofar software stack. Various aspects of the software stack are discussed:

- How to install the software stack
- An overview of the locations where the software stack is installed
- The structure of the software stack
- How to use the software stack
- An overview of all used libraries

The Lofar LTA software stack is the collection of software that is needed to run the Lofar imaging pipeline. That includes all needed libraries with a specific version. Different versions of the Lofar software can be installed at the same time.

For more information see the internal Lofar LTA Software Stack page (login needed).

Questions on installing the Lofar Software Stack can be send to the Software Support softwaresupport@astron.nl. Note that the build script is not supported outside of the scope of the LTA. Questions will be answered on a best effort basis. Requests for support or limited changes for installations that are available to multiple users (in particular on compute clusters) may be considered.

Build Script

The LTA pipeline build script is available from the Astro-WISE cvs repository; build_lofar_pipelines.py (username anonlta, password accesslta).

The build_lofar_pipelines.py script generates a shell script to do the actual build. For help do :

```
build_lofar_pipelines.py --help
```

To make a default build in directory lofar_build :

```
build_lofar_pipelines.py --root-dir lofar_build
```

and then run the generated script, this can take a few hours.

The generated shell script does the following:

- download all needed libraries
- build the libraries
- make a Lofar svn checkout (read only)
- build the Lofar package named "Offline"
- (not default) create tar with the build products

To initialize the environment variables to use the build, execute the following script in the build dir :

```
    init_env_release.sh
```

This script can also be used to build the Lofar LTA pipeline environment on worker or compute nodes of a Grid cluster. We used build_lofar_pipelines_on_grid_ce.sh as a wrapper of the build script to execute the build and store the results on a storage element. To submit a build as a grid job use build_lofar_pipelines_on_grid_ce.submit.sh, for example

```
    # create proxy certificate
    voms-proxy-init -voms lofar:/lofar/user -hours 48
    # submit build jobs to the grid environments
    env BUILD_LOCATION=rug.nl    BUILD_TIME=4
./build_lofar_pipelines_on_grid_ce.submit.sh
    env BUILD_LOCATION=sara.nl   BUILD_TIME=4
./build_lofar_pipelines_on_grid_ce.submit.sh
    env BUILD_LOCATION=nikhef.nl BUILD_TIME=240
./build_lofar_pipelines_on_grid_ce.submit.sh
```

Current LTA Installations

The table below gives an overview of the Lofar versions installed on the different compute environments of the LTA. The dates specify when the version was installed. An overview of all svn revisions can be found here https://svn.astron.nl/LOFAR/branches/

Lofar version	RuG (Grid)	RuG (millipede)	Sara (Grid)	Nikhef (Grid)	Jülich
LOFAR-Release-1_9	14-01-2013	10-01-2013	14-01-2013	14-01-2013	
LOFAR-Release-1_14	31-05-2013	31-05-2013	31-05-2013	31-05-2013	
LOFAR-Release-1_15	23-07-2013		23-07-2013	23-07-2013	
LOFAR-Release-1_16	04-10-2013		04-10-2013	04-10-2013	
LOFAR-Release-1_17	01-11-2013	01-11-2013	01-11-2013	01-11-2013	01-11-2013
LOFAR-Release-2_0	10-01-2014	10-01-2014	10-01-2014	14-01-2014	10-01-2014
LOFAR-Release-2_1	21-02-2014	21-02-2014	21-02-2014	21-02-2014	21-02-2014
LOFAR-Release-2_2		01-05-2014			
LOFAR-Release-2_3		28-05-2014			02-06-2014

Release notes can be found on the internal release notes wiki page. Any versions before 1.9 can be present on a compute environment but are not taken into account here and should probably not be used.

Layout and Structure

The Lofar LTA spans multiple (non CEP) compute clusters. To enable processing on all these clusters transparently to users and systems we'll define an uniform environment. The environment consists of the Lofar pipelines and related libraries. This section defines the directory structure of the Lofar pipelines and related libraries. This way we'll only need to define the Lofar root directory per cluster,

the directory structure below is the same for all clusters. On Grid systems the environment variable "VO_LOFAR_SW_DIR" is used to point to the pipeline environment. For simplicity we use this variable also for non Grid clusters.

The Lofar pipeline environment can contain multiple releases. One release consists of a certain version of the Lofar pipelines, including fixed versions of all the related libraries. From the Lofar root directory the directory layout is :

```
data
current <is link to specific version>
    lofar
    local
0.9
    lofar
    local
0.8
    lofar
    local
2011-07-12
    lofar
    local
... etc ...
```

The "data" directory contains measures data needed by the pipelines. This is static data during a pipeline run, although this directory should be updated regularly. The update mechanism needs to be determined. The contents of the data directory are provided by Astron at
ftp://ftp.astron.nl/outgoing/Measures/WSRT_Measures.ztar

The "current" entry is always a link to the latest release. Astron defines a release; the version of the Lofar pipelines and the related libraries. Next to the official Astron releases we can also support custom releases like nightly builds. In the above directory listing "0.9", "0.8" and "2011-07-12" are examples of releases.

The "lofar" directory contains the Lofar pipelines, everything that is compiled, build and installed from the Lofar sources using cmake. The "local" directory contains all the related libraries. The "local" directory behaves as the root directory for installing the libraries. Below is a detailed view of a release with the relevant sub directories shown

```
lofar
    release
            bin
        lib
        lib64
        ... more root dirs ...
    build_with_options
            bin
        lib
        lib64
        ... more root dirs ...
    ... more builds ...
local
```

```
    release
        bin
        lib
        lib64
        ... more root dirs ...
    build_with_options
        ... root dirs ...
... more builds ...
```

The Lofar source and libraries can be build with different options. The default location is "release" and should always be present.

Environment Variables

Before running the pipelines the following environment variables should be set to the given values. For readability the release "current" and build "release" are used. Ideally these environment variables should be set by the lofarinit.sh shell script. The problem is that this script sets these variables as they were during compile time, while this script can be deployed somewhere else.

Variable	Value
PATH	$VO_LOFAR_SW_DIR/current/lofar/release/bin
	$VO_LOFAR_SW_DIR/current/lofar/release/sbin
	$VO_LOFAR_SW_DIR/current/local/release/bin
LD_LIBRARY_PATH	$VO_LOFAR_SW_DIR/current/lofar/release/lib
	$VO_LOFAR_SW_DIR/current/lofar/release/lib64
	$VO_LOFAR_SW_DIR/current/local/release/lib
	$VO_LOFAR_SW_DIR/current/local/release/lib64
PYTHONPATH	$VO_LOFAR_SW_DIR/current/lofar/release/lib/python2.7/site-packages
	$VO_LOFAR_SW_DIR/current/local/release/lib/python2.7/site-packages

Note that :

- builds prior to version 1_14 used python 2.5
- from build 2_0 the local directory has release as sub directory

In the build directory are two scripts created to initialize the environment variables (source <script>):

- init_env_release.sh
- init_env_vo_release.sh

The first one initializes the environment variables for the environment where the build was done. The second one initializes the environment variables for a grid environment, where it is expected that VO_LOFAR_SW_DIR is defined.

Here is an example how to set the environment variables by hand in bash :

```
export
```

```
PATH=$VO_LOFAR_SW_DIR/current/lofar/release/bin:$VO_LOFAR_SW_DIR/current/lof
ar/release/sbin:$VO_LOFAR_SW_DIR/current/local/release/bin:$PATH
export
LD_LIBRARY_PATH=$VO_LOFAR_SW_DIR/current/lofar/release/lib:$VO_LOFAR_SW_DIR/
current/lofar/release/lib64:$VO_LOFAR_SW_DIR/current/local/release/lib:$VO_L
OFAR_SW_DIR/current/local/release/lib64:$LD_LIBRARY_PATH
export PYTHONPATH=$VO_LOFAR_SW_DIR/current/lofar/release/lib/python2.7/site-
packages:$VO_LOFAR_SW_DIR/current/local/release/lib/python2.7/site-
packages:$PYTHONPATH
```

Where $VO_LOFAR_SW_DIR should point to the installation directory.

External Libraries

The following libraries with given versions are installed in local directory in the LTA Pipeline
Environment by the build script.

Library	Version
astropy	0.2.4
bison	3.0
blitz	(removed)
boost	1.55.0
cmake	2.8.5
casacore	rev 21389
casarest	rev 8758
cfitsio	3240
fftw	3.2.2
flex	2.5.35
freetype	2.4.0
gsl	1.16
hdf5	1.8.4
h5py	2.2.0
libpng	1.5.6
libpqxx	4.0.1
libxml	2.9.1
log4cplus	1.0.4
m4	1.4.17
matplotlib	1.2.1
monetdb python client	11.16.0.5
numpy	1.7.1
OpenBLAS	0.2.6
PostgreSQL	9.1.2
pyrap	rev 345
python	2.7.5
scons	1.3.0
scipy	0.12.0

Library	Version
Twisted	13.1.0
setuptools	0.6c11
unittest-xml-reporting	1.7.0
wcslib	4.4.4
zlib	1.2.8
zope.interface	4.0.5

From:

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

Permanent link:

https://www.astron.nl/lofarwiki/doku.php?id=public:software_stack_installation&rev=1419000593

Last update: **2014-12-19 14:49**

Library	Version
Twisted	13.1.0