

Processing Non-Pulse-Based Observations

The backend used for CDMT is also available in a CLI, `lofar_udp_extractor`, which is installed on the [Docker containers](#) available on the REALTA nodes.

This guide assumes you have a UDP recording (compressed or uncompressed) from Olaf Wucknitz's VLBI recording program (standard for observing with I-LOFAR) and will explain the standard operating modes, and workarounds for issues with the `lofar_udp_extractor` program. [The full, up to date documentation for the CLI can be found here.](#)

Standard Usage

```
lofar_udp_extractor \  
-H /path/to/raw/udp_1613%d_TIMESTAMP.zst \  
-O /output/file/location \  
-p <procMode>
```

This sets up the program to take a compressed ZST file, starting at port 16130 and iterating up to port 16133, outputting to the provided location in a set processing mode. Some processing modes have multiple outputs, and will require '%d' to be in the output name as a result. The most useful processing modes are

Mode ID	Output (Stokes)	Tsamp (us)	Outputs
100	I	5.12	1
104	I	81.92	1
150	I, Q, U, V	5.12	4
154	I, Q, U, V	81.92	4
160	I, V	5.12	2
164	I, V	81.92	2

Modes 150+ are only available in more recent versions, and may error out of the docker containers have not been updated recently.

There are several other useful flags for processing data, such as `-u <num>` which will change the number of ports of data processed in a given run, `-t YYYY-MM-DDTHH:MM:SS -s <num>` or `-e <files>` can be used to extract a specific chunk of time, or specify a file with several time stamps and extraction duration (with the requirement that these regions do not overlap).

The `-s <flags>` flag passes flags to `mockHeader` which generates a sigproc-compatible header of metadata about the observation. This can make handling Stokes data easier later on, through the use of `sigpyproc` for loading and manipulating data, though as of right now it is not possible to set a per-subband frequency as is needed for mode357, so a dummy `fch1` (central top frequency) and `foff` (frequency offset between channels) should be used instead.

As an example, during a processing run on 29/10/20 of some Solar Mode357 data, the following command was used.

```
lofar_udp_extractor \  
{-H /mnt/ucc1_recording/data/sun/20201028_sun357/20201028090300Sun357/udp_1613%d.ucc1.2020-10-28T09:05:00.000.zst \  
{-o ./2020-10-28T09:05:00_mode357_StokesVector%d.fil \  
{-p 164 \  
{-a "-fch1 200 -fo -0.1953125 -tel 1916 -source sun_357"
```

Known Issues and Workarounds

When recording starts later than the supplied start time, Olaf's recorder may pick up stale packets in the UDP cache and record them at the start of your observation. This will manifest itself as a **segfault when trying to process the start of an observation**, as the program will run into issues attempting to align the first packet on each port. As a workaround, use the `-t YYYY-MM-DDTHH:MM:SS` flag to set a start time shortly after the actual data begins recording, at which point the software will be able to accurately align the packets as needed,

Revision #3

Created 29 October 2020 16:33:01 by David

Updated 6 November 2020 14:03:09 by David