

DiFX Exercises for the 2016 IVS school

Notation and basics

In the documentation that follows, things that you type will typically be in fixed-width type like this.

It is assumed here that the student has some knowledge of Linux/Unix and the bash shell, though most of the demo does not really need much knowledge. Basic knowledge of editing text files will be needed. If help is needed with that, please ask for assistance early.

Pretty much every one of the programs that you will use in this demo has built in help information which can be accessed by running at the command line with the -h option. For example:

```
vex2difx -h
```

In general it would be good to run this for each unfamiliar program to get an idea of what options are available and for help on syntax.

If more detailed help is needed, the [Reference Manual](#) is the first place to go. Note that the information in the manual is not always up to date. Comparing the help information of a program (with the -h option) and the description in the manual will give you an idea if the manual is up to date or not.

Preliminary

DiFX is a suite of programs, libraries and small utilities that is generally installed in a non-system directory, making it inaccessible to the shell without some configuration. An actual production correlator would likely have multiple versions of the correlator installed. Usually each one would be selected through its own setup script. For the existing demo a single DiFX version is available and can be configured by typing

```
setup_difx
```

at the command prompt. This command will need to be entered separately for each shell that is started (i.e., for each terminal window opened).

Start the delay model server

The delay model server needs to be started a single time after the computer is started. This can be done with:

```
startCalcServer
```

This starts in the background a process called CalcServer.

Dataset 1: 2-station UT1-UTC observation

Create your work directory

Make a new directory in your home directory. Call it whatever you like. In this documentation it will be called dataset1. Then enter that directory.

```
mkdir dataset1
```

```
cd dataset1
```

Copy the .vex file and nominal .v2d file

The .vex file is generated by the experiment scheduling program and contains the basic description of the experiment, the coordinates of sources observed, and the setup used in each scan. Most of the information needed to correlate an experiment is contained in this file.

```
cp /home/avntrainee/difx_data/n6043/n6043.vex.preobs .
```

The .v2d file supplements the .vex file. It contains two types of information: 1. information that is not contained in the .vex file, such as the integration time (also called accumulation period), and the spectral resolution to be used at correlation; and 2. information that overrides values contained in the .vex file. The general philosophy is that the .vex file should not need editing and the changes needed to correlate should be contained in the .v2d file.

```
cp /home/avntrainee/difx_data/n6043/n6043.v2d .
```

The name of the .v2d file (everything before the .v2d extension) forms the base of the correlator job names. In order to prevent confusion (more on this later) please rename the .v2d file to something unique. This could be your name or the city where you come from.

```
mv n6043.v2d walter.v2d
```

In the examples that follow, I'm assuming it is called walter.v2d. Please replace my name with yours.

Generate listing of baseband data to correlate

The correlator needs to know where to find data and what time ranges to associate with each file. This can be done with tools that inspect the data itself. For this particular experiment the data were recorded with a data format called Mark5B (see http://www.haystack.mit.edu/tech/vlbi/mark5/mark5_memos/019.pdf). A tool called m5bsum can tabulate the data. Do this separately for data from each of the two stations (PT = Pie Town, and MK = Mauna Kea, both are VLBA antennas in the United States).

```
m5bsum -s /home/avntrainee/difx_data/n6043/NRA0+301* > filelist.pt
```

```
m5bsum -s /home/avntrainee/difx_data/n6043/NRA0+390* > filelist.mk
```

The `-s` option is important. This forces generation of a one-line summary for each file that can be interpreted by later software. Note that you can leave these large data files where they currently are. If you are curious to see more details for a single file you can try:

```
m5bsum /home/avntrainee/difx_data/n6043/NRA0+390_0009_N6043_MK_No0010
```

The two files created here will need to be referenced in the `.v2d` file in preparation for correlation. The files are ASCII text and can be viewed with `cat`, `less`, or your favorite editor. The long numbers are start and stop time expressed as Modified Julian Days.

```
cat pt.filelist
```

Inspect and possibly modify .v2d file

The `.v2d` file was already created for you. Many things in the file can be changed. For a first correlation pass I suggest not changing anything. If time permits it might be interesting to change some parameters. If you do this, it might be good to rename the `.v2d` file so you can preserve output from previous correlations. Some relevant parameters to try are:

- `tInt` Accumulation Period (seconds): good numbers are between 0.25 and 4
- `specRes` Spectral resolution (MHz) of correlator output. Good numbers may be between 0.05 and 0.5
- `fftSpecRes` The resolution at which the FFT is performed. Must be a `specRes` must be an integer multiple. This affects the spectral response.
- `clockOffset` Change the clock offset (microseconds) and incur a phase slope in output. Try numbers up to 0.2 or so.
- `addZoomFreq` Allow zooming into a portion of a band. This is an advanced feature – probably want to ask for help.

Note that fairly complete documentation for `vex2difx` can be found at [vex2difx](#).

Once the `.v2d` file looks OK, run `vex2difx`:

```
vex2difx walter.v2d
```

Look carefully at the output. There may be meaningful messages. If you see an Error, then the step failed and you will need to debug your `.v2d` file. If you see a Warning, things might be OK, but look carefully at the message as it might mean something is suspicious. If you see a Note, then some assumption was made within `vex2difx` that is probably expected, but you might want to make sure it makes sense.

`vex2difx` generates one or more job for a project. Each job has a different trailing number (after an underscore). Each job at this stage should already have a few files. The `.input` file contains the DiFX configuration for the job and the `.calc` file contains information needed to generate the delay polynomials. Both of these files are ASCII text and can be viewed as was done for the filelist files.

Generate delay polynomials

The delay polynomials are used to delay each datastream by the amount of time needed for the wavefront to travel from the received station to the Earth center, a number typically 10 milliseconds

or less. The Goddard Calc program is used to do this. Current installations use Calc 9.1 (via the instructions here), but Calc 11 is available to difx through a program called difxcalc (see Poster by David Gordan at the subsequent IVS school). The simplest way to execute Calc is to run

```
calcif2 -a
```

which will go through all .calc files in the current directory and produce model files (ending in .im). Again these output files are ASCII text and can be examined.

Start some monitor processes

As DiFX runs it produces informational (sometimes error) messages. It also generates status information. These bits of information are multicast out for any process on the local network to hear. This means you will be able to see messages coming from your neighbor's processes (could either be a feature or a nuisance). To filter messages that are relevant only to your process, use the standard unix tool grep to filter the output. If you want you can run without grep and see everybody's information flowing at once! Note you can start and stop these monitor processes even while correlation is in progress.

Start the error/message monitor

Start a new terminal window and in the shell set up the environment:

```
setup_difx
```

Then run

```
errormon | grep walter
```

But don't forget to change from my name to the name of your process.

Start the status monitor

Start a new terminal window and in the shell set up the environment:

```
setup_difx
```

Then run

```
statemon | grep walter
```

Run DiFX!

At this point all the files for one or more jobs should be ready to correlate. Without further explanation, DiFX can be run with the following command:

```
startdifx -v -f -n walter*.input
```

Note that the `-f` option forces execution and will overwrite previous correlation results for the same job. Use with care! The `-v` option simply increases verbosity of the output, and the `-n` option is needed in this case to prevent overwriting the machines file – a file describing which elements of the computer cluster (in this case just your workstation) to use.

Running these jobs will take about 10 minutes. While it is running you can monitor the two diagnostic windows. When that gets boring you can try some [additional things](#).

Note that data for only a subset of the scans for this project are available.

Convert to FITS output

The program `difx2fits` takes the raw output from DiFX and the files used to drive it to generate a file in FITS-IDI format. FITS-IDI is a standard format for interferometry data and is documented at <http://www.aoc.nrao.edu/~egreisen/AIPSMEM114.PDF>. This file format is suitable for reading into AIPS, the usual data path for astronomical VLBI observing. The FITS file is not something to be explored at this school.

`difx2fits` also generates some diagnostic output which can be very handy (see next section).

To run `difx2fits`:

```
difx2fits walter*.difx WALTER.FITS
```

Make diagnostic plots and inspect

A tool called `difxsniff` can be used to make “sniffer plots” for quick look assessment of the data. These plots are used at NRAO by data analysts to determine the success of a project. To run:

```
difxsniff WALTER.FITS PT
```

The second argument, `PT`, is the station which will serve as the “reference antenna” in some of the plots. Usually a well-behaved antenna near the center of the VLBI array is used for this. This will generate files in a subdirectory called `sniffer/PT/`. The following files will be of primary interest:

- `WALTER.apd.ps` : plot showing Rates, Delays, Phases and Amplitudes on all baselines to the reference antenna as a function of time
- `WALTER.acb.ps` : plots of all auto-correlation bandpasses
- `WALTER.xcb.ps` : plots of all cross-correlation bandpasses to the reference antenna
- `WALTER.wts.ps` : plots of data validity weights as function of time

All of the postscript files (files ending in `.ps`) can be viewed with `evince`, e.g.,:

```
evince WALTER.apd.ps
```

Make Mark4 output for geodetic processing

Making sure to be back in the experiment directory (i.e., if you went into the sniffer directory, come back out), you can make a Mark4 data set from the same DiFX output. This can be invoked with the

following:

```
difx2mark4 walter*.input
```

The output files will be put in an numbered directory (default is 1234/). Roger Cappallo can explain what is done with these files at this point.

Dataset 2: 10-station VLBA observation (if time permits)

Using the tools above (with some exceptions: see below), try to correlate a 10 station VLBA observation. The data in question make use of multi-threaded VDIF data (see http://vlbi.org/vdif/docs/VDIF_specification_Release_1.1.1.pdf if you are curious how VDIF data is formatted). Below is some minimal information that should allow you to get started:

1. A single scan from this experiment is available.
2. Data is from a 12.2 GHz observation of a quasar with 4 baseband channels, each 4 MHz.
3. The baseband channels come in polarization pairs: RCP and LCP.
4. Data files can be found at /home/avntrainee/difx_data/ts036h (station name is second to last field of filename)
5. Vex and sample .v2d files can be found in the same place (called ts036h.vex.obs and ts036h.v2d)
6. To generate the file lists, you need to use `vsum -s` instead of `m5bsum -s`
7. You will need to make 10 file lists. Note that the station codes for the 10 VLBA stations are: BR, FD, HN, KP, LA, MK, NL, OV, PT, SC, as listed at the top of the .v2d file.

From:

<http://www.atnf.csiro.au/vlbi/dokuwiki/> - **ATNF VLBI Wiki**

Permanent link:

<http://www.atnf.csiro.au/vlbi/dokuwiki/doku.php/difx/ivs2016?rev=1457215647>

Last update: **2016/03/06 09:07**

