VLBA-DiFX Operations Plan

VLBA Sensitivity Upgrade MEMO 25

Walter Brisken

National Radio Astronomy Observatory

February 18, 2009

1 Introduction

Migrating from the hardware correlator to the DiFX software correlator will require some moderate changes to the way jobs are prepared and data is sent to the archive. A major philosophical change is the break from purely *job based* processing to a system where jobs stand alone only in the correlation portion of the processing. Changes to the system also include naming of correlator passes, integration of sniffing with the generation of FITS files, and the complete bypass of DAT tape except when DAT is requested as the distribution media. Despite limited software development resources the new operations system should be better streamlined than the existing system and should require far less hand-editing of files. In this document, a hypothetical pulsar astrometry project is used to illustrate a particularly complicated case. Most projects will be substantially simpler. The form the operations plan takes will have no effect on the accuracy of the results of correlation so the implementation of this plan can proceed in parallel with the planned tests of the software correlator.

2 VLBA project life cycle

Under the proposed operational plan, the life stages of a project are as follows. Figure 1 diagrams the software involved in the correlation process.

1. *(observer)* observation `.key` file is submitted.
2. *(analyst)* `sched` generates the observe vex file. Currently the vex file ends with `.skd` which is kind of confusing. For now we should live with this, but when convenient change to `.vex`.
3. *(operators)* Project observes.
4. *(operators)* `TSM` is run to generate the `cal.vlba` file.
5. *(automatic)* `mod2db` is run to get monitor data into database. Eventually `mon2db` and `fs2db` will be supplemented and/or replaced with a new program that can handle the new Digital Back-end (DBE) and Mark5C data.
6. *(analyst)* `vlog` is run on the cal.vlba file to format information that will be needed for `difx2fits`.
7. Wait for modules to arrive in Socorro and for Earth Orientation Parameters (EOPs) to become final (if requested by observer).

8. *(analyst)* `db2vex` is used to extract clock offset, EOPs, and disk module information from the database to form the *observed* vex file (currently ending with .skd.obs).
9. *(analyst)* One `.v2d` file is made (see § 3) for each correlator pass that is needed. A program `oms2v2d` exists that makes a starter `.v2d` file from the `.oms` file that sched creates.
10. *(analyst)* `vex2difx` is run to produce DiFX input files. These files include those ending in .input, .calc, and .flag. Additional files may be generated in the future. `vex2difx` is run separately for each correlator pass.
11. *(analyst)* `difxqueue` is used to copy DiFX input files to the software correlator. At this time a row is written to the database for each job indicating that it is ready for correlation. As part of this step the delay model is calculated for each job using `calcif2`.
12. *(operators)* Correlation is performed. The GUI will find jobs to run from the database and will write a row to the database indicating completion.
13. *(analyst)* FITS files are created using `difx2fits`. `difx2fits` is run separately for each correlator pass. Sniffer data files are created during this process. It is at this step that transfer of calibration data is performed.
14. *(analyst)* `difxsniff` is used to produce the sniffer plots which are subsequently analyzed.
15. *(analyst)* `difxarch` is used to copy data from the DiFX staging area into the NRAO data archive.
16. *(analyst)* Module resources for project are released.
17. *(observer)* Once archiving is complete the data can be downloaded.

DAT tapes are no longer needed in the archiving process meaning getting data into the archive should be much simpler. The same project correlated on DiFX will require only about 40% of the archive storage area as the equivalent output from the hardware correlator as a result of not needing to run the VLBA pipeline on the FITS files. The output FITS files from DiFX will already have calibration transfer applied and will not require sorting or splitting into different output files by frequency. Distribution of data will be done entirely from the archive. If a user requests a DAT tape, it will have to be copied from the archive to DAT locally by operations; hopefully this will be rare.

3 Correlator jobs

The *job* is the fundamental unit of correlation in both the old (hardware – hereafter FXCORR) and new (software – VLBA-DiFX) correlators. It is often the case that a single project needs to be split into many jobs. There are several reasons for splitting jobs which are summarized in Table 1. Effort is being made to minimize the number of reasons to split jobs. For both correlators there may be practical reasons to split a project into more than the minimum number of jobs. Jobs with run-times of more than a few hours are currently avoided to prevent total loss of progress in the case of failure during correlation, although this is not a fundamental limitation of either VLBA-DiFX or FXCORR.

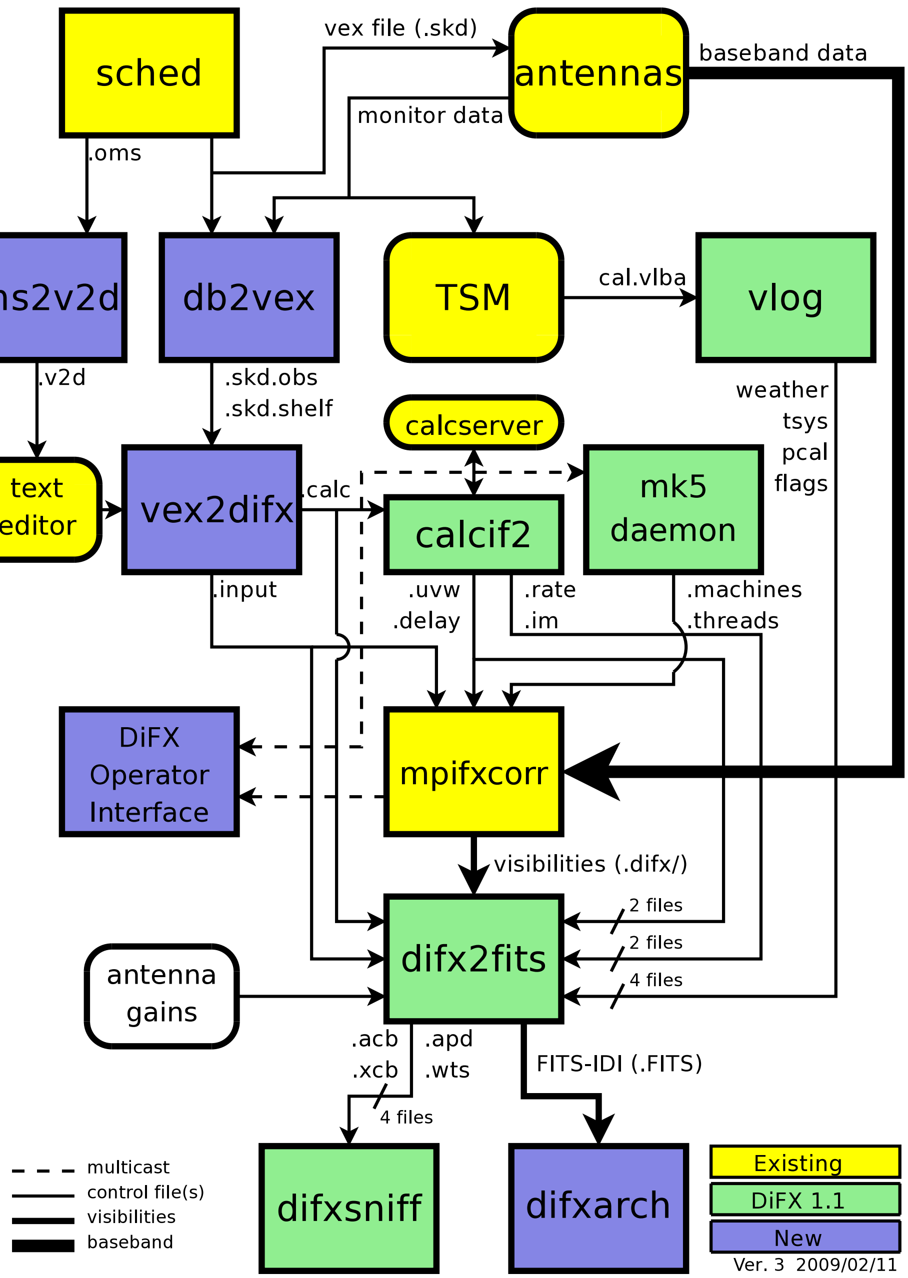

Figure 1: *Data flow for vex-based correlation* Note that some of the programs are will not be explicitly run, but may rather be started though another program. Entities represented by boxes with rounded corners are to be used by VLBA-DiFX without any modification.

FXCORR uses a numbering system to identify the multiple jobs for a project. The job number has two parts: the series (1 or 2 digits) and the job number within the series (2 digits). Some conventions are followed in the numbering of jobs. Single-digit series are test projects. Job numbers less than 20 are test correlations, usually clock searches. While this system has been working for

Reason	FXCORR	VLBA-DiFX(now)	VLBA-DiFX(later)
Midnight crossing	Y	N	N
Output file size	Y	N	N
More than 8 sub-bands	Y	N	N
More than 32 tracks	Y	N	N
Multiple pulsar gate phases	Y	N	N
Media Change	Y	Y	?
FITS-compatibility	Y	Y	?
Multiple phase centers	Y	Y	?
Multiple sub-arrays	N	Y	?

Table 1: List of reasons that a project must be split into multiple jobs. Some of the requirements on splitting jobs may be lifted in future VLBA-DiFX versions.

the last 15 years, it stands to be improved. Major short-comings include limit to the number of production jobs (80) and loss of intent in the naming scheme (i.e., can't tell which pass of a multiple-pass project a particular job refers to). VLBA-DiFX has another characteristic that will benefit from a new job naming scheme; the one-to-one mapping of correlator jobs to FITS files is no longer a strict correspondence. A single job can contain multiple frequency settings (AIPS frequency IDs) that logically belong separated; `difx2fits` will separate different frequencies into different FITS files. Additionally, multiple correlator passes that were broken for reasons such as media change can be merged into a single FITS file. These changes make for a more practical archiving and use of data but would result in significant confusion if the naming convention were not changed. Note that a proof-of-concept operational system for DiFX at the VLBA correlator was built around the correlator jobs written for FXCORR and thus inherited conventions from the hardware correlator. This document discusses a production-ready operational system that depends only lightly on software infrastructure designed for FXCORR; all software involved in the creation of correlator jobs and involved in steps that logically follow is specifically designed for use within the VLBA-DiFX system.

The proposed operation plan makes extensive use of the concept of a *correlator pass*. In this context a correlator pass is a group of one or more jobs that are naturally grouped together in correlation preparation and archiving. A pass can comprise of an entire project or a portion thereof. No two jobs of a given pass can specify correlation of a given antenna for the same time; if such functionality is needed an additional pass will be required. It is however possible that two jobs are interleaved in time. This is most likely to happen in certain fast-switching experiments where the two sources cannot be correlated with the same `.input` file. These job passes are explicitly named giving explicit intent to each pass. Examples of passes would include clock trials, correlation of the geodetic blocks of an astrometry program, or one of several passes to accommodate multiple phase centers. The jobs in a pass are numbered starting at 1 to allow the job number 0 to be special in the future if deemed necessary. Due to the nature of sub-arraying and the algorithm used to generate the jobs of a pass it is not necessarily the case that jobs with higher sequence numbers are from later parts of experiments, but this may change. Leading zeros will be prepended to the job number to ensure proper alpha-numeric ordering of jobs within a pass. A complicated (perhaps even contrived) example might have a list of jobs similar to those listed in Table 2. Each pass has its jobs derived from a single `.v2d` file. The jobs for this hypothetical project may be generated by the four `.v2d` files shown in Sample files 1-4. Correlation of a project will typically proceed in time order, even if that means starting the correlation of a second pass before finishing the first. Exceptions may be cases where certain jobs are submitted with elevated priority which would override the time ordering. Aiming for time-ordered correlation will minimize module swapping at the correlator and streamline iteration through the correlator queue.

Job name	Start time	Stop time	Comments
clock1	10:00:00	10:05:00	observation of a bright calibrator
geodetic1	10:00:00	10:26:43	first block of geodesy calibrators
geodetic2	10:26:43	10:40:00	job break due to media change
geodetic3	12:00:00	12:40:00	second geodetic block
geodetic4	14:00:00	14:40:00	final geodetic block
inbeam1	10:40:00	12:00:00	first block of science data
inbeam2	12:40:00	13:34:12	second block of science data
inbeam3	13:34:12	14:00:00	another media change
pulsar1	10:40:00	12:00:00	first block of science data different coordinate pass
pulsar2	12:40:00	13:34:12	second block of science data
pulsar3	13:34:12	14:00:00	media change

Table 2: Example application of pass names to correlator jobs for a complicated example.

Sample file 1 *clock.v2d* – `vex2difx` file for generation of clock search jobs. Generation of the clock job is done at the command line with command `vex2difx clock.v2d` .

```
vex = bx123.skd.obs
jobSeries = clock

SETUP clock { nChan=256 tInt=1 }
RULE clock { source=DA213 setup=clock }
```

4 Step-by-step guide to correlating

This final section details the commands the analyst will use in the preparation of correlator jobs through the archiving of data. The actions of the correlator operator will be detailed in a separate document. The particular case being described here is based on the complicated pulsar astrometry project that has been exemplified above. Most real-life examples will be simpler, but some may be more complex. Note that these instructions represent the expected way to proceed, but changes to the software architecture may introduces changes to some of these steps.

It should be kept in mind that all actions performed by the analysts will be *pass based* which means one or more jobs at a time. Rarely will analysts have to worry about individual jobs or FITS files. The correlator operators on the other hand work entirely on the job basis. Commands to be issued by the analysts are preceded by an arrow ($\longrightarrow$).

Sample file 2 *geodetic.v2d* – `vex2difx` file for generation of geodetic block jobs. This assumes that the setup name used in the sched `.key` file is called "geodetic". Since no "default" SETUP was provided only scans using mode "geodetic" will be correlated.

```
vex = bx123.skd.obs
jobSeries = geodetic

SETUP geodetic { tInt=2 nChan=32 }
RULE geodetic { mode=geodetic setup=geodetic }
```

Sample file 3 *inbeam.v2d* – `vex2difx` file for generation of inbeam calibrator block jobs. The RULE section specifies that any scan using mode "geodetic" is to be ignored. All other scans are to be correlated with 2 second integrations and 64 channels per sub-band.

```
vex = bx123.skd.obs
jobSeries = inbeam

SOURCE B0329+54 { ra=03:34:34.034 dec=54:38:21.12 calCode=T }

SETUP default { tInt=2 nChan=64 }
RULE geodetic { mode=geodetic setup=SKIP }
```

Sample file 4 *pulsar.v2d* – `vex2difx` file for generation of gated pulsar pass. Note that only the pulsar is correlated here.

```
vex = bx123.skd.obs
jobSeries = pulsar

SOURCE B0329+54
{
  ra=03:32:56.034
  dec=54:10:22.12
  calCode=G
  polyco=B0329+54.polyco
}

SETUP pulsar { tInt=2 nChan=64 }
RULE pulsar { source=B0329+54 setup=pulsar }
```

1. First change to the project directory. Assume that the project is called BX123 and that it was observed in December 2009.

   ⟶ `cd /home/vlbiobs/astronomy/dec09/bx123`

2. Extract from the monitor database the Mark5 module logs, clock offsets and rates, and EOPs making a new vex file called `bx123.skd.obs` and a file called `bx123.skd.shelf`. The original (schedule) vex file that was used during observation is never to be modified. In extreme cases, the new vex file being created in this step `bx123.skd.obs` can be hand edited to reflect what actually happened during observation, but doing this should be extremely rare. This step locks in the EOP values that will be used for each job made in this project.

   ⟶ `db2vex bx123.skd`

3. Next form the template input file for `vex2difx` from the `.oms` file written by `sched` . This creates `bx123.v2d` .

   ⟶ `oms2v2d bx123.oms`

4. For simple experiments it is likely that the `.v2d` file created in the previous step can be used unmodified. For this complicated experiment changes will need to be made. Since this project requires four correlator *passes*, this `.v2d` file will need to be copied four times and each one edited to reflect the purpose of the correlator pass. Sophisticated VLBA users may provide their own set of `.v2d` files that might need light editing before use.

   ⟶ `cp bx123.v2d clock.v2d`

   ⟶ `emacs clock.v2d`

5. VLBA-DiFX `.input` files are generated at this point using `vex2difx`. By design, `vex2difx` has no options associated with it – it is entirely configured through the `.v2d` files. In the case below, the files `clock1.input`, `clock1.calc`, and `clock1.flag` will be created. This command will also make a file called `clock.joblist` (see Sample file 5) that lists each job created for this correlator pass with a summary of the job properties, such as start and stop times and number of stations.

   ⟶ `vex2difx clock.v2d`

6. If the correlator jobs created above are deemed ready to run, they are sent to the correlator queue. In this process three things will occur: 1. `CALC` will be run to generate the correlator delay models needed for correlation, 2. the `.input` files generated by `vex2difx` will be copied to the software correlator run directory, and 3. the VLBA database will be told that the jobs are ready. At this time, a priority can be set to the jobs being sent to the correlator, making them appear at the top of the queue. Otherwise the jobs in the queue will appear in observe time order. In the example below, the option `-p 1` indicates that this job should run with elevated priority. Supplying `clock` with no prefix implies queuing all the jobs in the clock pass. Individual jobs could be queued by specifying a list of `.input` files.

   ⟶ `difxqueue -p 1 clock`

7. When the jobs are complete, which can be determined with `difxqueue` using the `-s` option, the correlator output is converted to FITS format. Data "sniffing" happens automatically

during this step. The command to do this will ensure that all of the jobs in the pass have been successfully correlated. Note that the number of FITS files created is not necessarily the same as the number of correlator jobs. A file called `clock.fitslist` (see Sample file 6) will be generated in this step that lists all of the fits files that are part of this correlator pass including for each FITS file a list of the jobs that contributed to that FITS file. The program `makefits` will use program `difx2fits` to do the actual conversion.

⟶ `makefits clock.joblist`

8. The sniffer output files are at this point inspected. Program `difxsniff` is run to produce plots which are identical to those produced by sniffer today. Multiple reference antennas (in this example, Los Alamos and Kitt Peak) can be provided at the same time. Sniffer plots and the data that is used to generate them will be placed in a sub-directory of the project directory called `sniffer/clock` for a pass called "clock".

   ⟶ `difxsniff LA KP clock.fitslist`

   ⟶ `gv sniffer/clock/apdfile.ps`

9. If the FITS files are deemed acceptable, they are interred into the VLBA data archive.

   ⟶ `difxarch clock.fitslist`

Sample file 5 *clock.joblist* – file listing all of the jobs (just one here) in the pass called "clock". The first line of a `.joblist` file contains a key=value list that can in principle contain arbitrary information. Here are listed five useful bits of information: the v2d file used to produce the jobs, the name of the correlator pass, the Modified Julian Day (MJD) that `vex2difx` was run, the version of DiFX being used, and the version of `vex2difx` that was used. All MJDs used in this document should be assumed to be UTC days and include the fractional day encoding the time since midnight. Each additional line of a `.joblist` file contains information for one of the jobs. Four fields are listed for each job: the name of the job, the start MJD of the job, the stop MJD of the job, and the number of antennas. Comments in this file type start with a #. Here a comment is used to list the antennas in the job.

```
v2d=clock.v2d  pass=clock  mjd=55187.666667  DiFX=DiFX-1.5  vex2difx=0.3
clock1 55180.416667 55180.420139 4  # KP LA OV MK
```

---

5 Project file management

Many files will be generated for each project, some of which will need to be preserved and some that will not. In general, all files that are made and stored in the `/home/vlbiobs/astronomy/` directory structure will be kept indefinitely. This includes all the files used at observe time, VLBA-DiFX `.input`, `.calc`, and `.flag files`, summary files for the correlator passes *pass*`.joblist` and *pass*`.fitslist` files, and any files produced by sniffer. Sub-directories for each correlator pass will be made to contain the job-specific files for each pass. The `.joblist` and `.fitslist` files for each pass will live in the project root directory. The raw correlator output and the FITS files that are formed will be temporarily stored on the software correlator head node until the data are

Sample file 6 *clock.fitslist* – file listing all of the fits files (just one here) in the pass called "clock". The first line of a `.fitslist` file contains a key=value list like that of the `.joblist` file. The information stored in this example includes the `.joblist` file associated with this pass, the name of the pass, the date when fits files were created, and relevant version numbers. There will be an additional line for each FITS file that is produced. Each of these lines consists of the name of the FITS file that was produced, the size (in units of 10^6 bytes) of the FITS file, and a list (here only 1 long) of jobs that contributed to the FITS file.

```
jobs=clock.joblist  pass=clock  mjd=55188.139965  DiFX=DiFX-1.5  difx2fits=0.8
VLBA_BX123_clock_1_091223T032133.idifits 12.5 clock1
```

archived. Periodically the software correlator staging area where these files are stored will need to be manually purged of old project data that was never archived. Special provisions for handling extremely large (hundreds of GB) correlated data sets will need to be made on a project-by-project basis.

6 The VLBA database

Many of the VLBA tools (such as the Observation Management System (OMS), `mon2db`, `cjobgen`, and others) make use of an Oracle database for persistent storage of various information related to projects that use either the VLBA antennas or correlator. Many aspects of VLBA-DiFX are not a good match for the existing database tables; adapting the existing tables to work nicely with VLBA-DiFX will be disruptive and have implications for much existing code, including software that will not be needed once FXCORR is shut down. The proposed solution to this dilemma is to use a parallel set of database tables for correlation and archiving when using VLBA-DiFX. The use of existing software for generation of FXCORR jobs will continue unchanged. For projects to be correlated using VLBA-DiFX, `OMS` will still be used for observation preparation tasks, but will not be used in preparation of correlation or anything that occurs beyond that in the project's life cycle. Instead, `vex2difx` will be used to generate jobs, `difxqueue` will be used in lieu of `OMS` to stage correlator jobs, and `difxarch` will be used in the archiving of data. The queuing tool `difxqueue` will be used to display the state of the VLBA-DiFX job queue as well as populate it. The new tools will access three new database tables: DIFXQUEUE and DIFXLOG; the contents of these tables is shown in Tables 3 & 4.

7 Archiving

Archiving of VLBA-DiFX data will be done on a per-pass basis. All `.FITS` files associated with a single correlator pass will be archived together. A particular staging directory for VLBA-DiFX data has been set up. Populating the archive amounts to first copying the files to be archived to this directory making sure that the first character of the file name is ".". Once the entire file is transferred this file is renamed without the leading period. This system is the standard way to populate the Next Generation Archive System (NGAS)[1] without potential for an incompletely

[1] see `http://www.eso.org/projects/dfs/dfs-shared/web/ngas/`

Column	Type	Comments
PROPOSAL	VARCHAR2(8)	The proposal code
SEGMENT	VARCHAR2(4)	Segment (epoch) of proposal, or blank
JOB_PASS	VARCHAR2(32)	Name of correlator pass (e.g. "geodesy")
JOB_NUMBER	INT	Number of job in the pass
PRIORITY	INT	Number indicating the priority of the job in the queue 1 is highest
JOB_START	DATE	Observe time of job start
JOB_STOP	DATE	Observe time of job stop
INPUT_FILE	VARCHAR2(256)	Full path of the VLBA-DiFX input file
STATUS	VARCHAR2(32)	Status of the job, perhaps "QUEUED", "KILLED", "RUNNING", or "COMPLETE"
NUM_ANT	INT	Number of antennas in the job

Table 3: The DIFXQUEUE database table. This table is based on the FXQUEUE table currently used by `OMS`. Entries to this table will be initially made by `difxqueue`. The STATUS field will be automatically updated as appropriate during correlation.

copied file to be archived. The file names will be composed only of alpha-numeric characters and "." and "_". These characters have no special meaning in any relevant software, including http, XML, bash/Linux command lines, the oracle database parser, etc. File names will have the following format:

`VLBA_`*projectCode*_*passName*_*fileNum*_*corrDate*`T`*corrTime*`.idifits`

where the italicized fields, which themselves will be limited to alphanumeric characters, are as follows:

Field	Type	Comment
projectCode	string	Project code, including segment if appropriate
passName	string	Name of the pass, as set in the `.v2d` file
fileNum	integer	FITS file sequence number within pass
corrDate	date (*yymmdd*)	Date corresponding to correlation completion
corrTime	time (*hhmmss*)	Time corresponding to correlation completion

Parameter *fileNum* is the sequence number of the created `.FITS` file which may or may not have a direct correspondence with the job sequence number within the correlator pass. An example archive file name relevant to the sample project used in this memo may be:

`VLBA_BX123_clock_1_091223T032133.idifits`

All files produced for a given pass will be placed in a single directory,

`$NGAS_ROOT/`*projectCode*`/`*passName*

where `NGAS_ROOT/` is an environment variable pointing to the head of the archive staging area for VLBA-DiFX. During the transfer to the archive, the *projectCode* portion of the directory tree will begin with a period that is to be renamed once all files are completely copied. This will allow the

Column	Type	Comments
PROPOSAL	VARCHAR2(8)	The proposal code
SEGMENT	VARCHAR2(4)	Segment (epoch) of proposal, or blank
JOB_PASS	VARCHAR2(32)	Name of correlator pass (e.g. "geodesy")
JOB_NUMBER	INT	Number of job in the pass
CORR_START	DATE	Start time/date of correlation
CORR_STOP	DATE	Stop time/date of correlation
OUTPUT_FILE	VARCHAR2(256)	File name of correlator output
OUTPUT_SIZE	INT	Size (in 10^6 bytes) of correlator output
CORR_STATUS	VARCHAR2(32)	Status of correlation, typically "COMPLETED"

Table 4: The DIFXLOG database table. This table is based on the FXLOG table currently used by `OMS`. A row will be written to this table after each successful correlation by the DiFX Operator Interface.

archive loader to logically group together all the files of the pass. If needed, an index file listing the association of archive `.FITS` files and correlator jobs can also be placed in this directory. In order to ensure the atomic nature of correlator passes in the archive, the renaming of the copied files from the temporary versions starting with "." will not occur until all archive files are transferred. The `.fitslist` file produced by `difx2fits` would serve this purpose. An archive loader will periodically (initially about every 30 minutes, but perhaps later with much shorter intervals) look for new files in the archive staging area to store. The archive data will be available moments later for users wanting to download the data.

8 Software development plan

The list below contains all of the required software pieces, describing their current status and estimating how much work remains to meet the needs of this operational plan.

1. `vlog` This is a small script that has existed for some time and is deemed stable with no required changes. The name of this program is based on the AIPS task of the same name and function which is to read the output of `TSM` and produce several calibration files. The output formats for the AIPS task and this script are not the same.

2. `db2vex` This program reads from several database tables to populate the `CLOCK`, `EOP` and `TAPE_LOG` vex tables. No changes are required at this point and documentation is complete.

3. `oms2v2d` This program reads the `.oms` file written by sched to produce a template `.v2d` file needed by `vex2difx`. Documentation is complete and no changes are required at this point. Note that none of `vlog`, `db2vex`, and `oms2v2d` require any user input and could be wrapped into a single script to simplify usage by analysts.

4. `vex2difx` This is a complex program that converts a vex file into one or more VLBA-DiFX input files. This program is mostly complete and is reasonably well documented[2] at this

[2]see `http://cira.ivec.org/dokuwiki/doku.php/difx/vex2difx`

point. Only minor work, including the generation of the `.joblist` file, is required to meet the requirements of this operations plan. A more substantial amount of work that can be more leisurely spread over the next few months will be needed to meet the goals of the `vex2difx` program. Time estimate to integrate with this operations plan: **4 hours**. Time estimate to completion: **10 days WFB**.

5. `difxqueue` This program will allow jobs to be added to the correlator queue and for the queue to be queried. Work has not yet begun. Time estimate: **3 days WFB**

6. `DOI` The Difx Operator Interface (DOI) will require moderate changes for its interface with the VLBA database. Time estimate: **1 month? MG**

7. `difx2fits` Conversion of raw VLBA-DiFX output (in native *difx* format) to FITS format files is done using `difx2fits`. This program is mature and is documented. Remaining work includes writing the `.fitslist` file and improving the heuristics of the built in "sniffer". Time estimate: **3 days WFB**

8. `makefits` This script will simplify the use of `difx2fits` and will abstract away the location of the software correlator data staging area. Work has not yet begun. Time estimate: **1 day WFB**

9. `difxsniff` Minor changes will be needed to teach `difxsniff` to look for files on the software correlator head node and place the sniffer data files and plots in the `/home/vlbiobs/astronomy/` project directory. Time estimate: **2 hours WFB, 1 day RCW**.

10. `difxarch` This program will copy data to the VLBA data archive staging area and will clean up temporary files on the software correlator temporary data area. Work has not yet begun. Time estimate: **1 day WFB**

9 Analyst GUIs?

The initial suite of software that will fulfill the needs of correlation preparation and archiving will not include a graphical user interface (GUI). The first priority is to develop the command line tools described here. GUI wrappers for some or all of this functionality will be the second priority. It is plausible that eventually GUIs will enable all of the functionality to be performed, however it should be understood that properly designed GUIs require a significant effort. GUIs will likely be developed in stages; feedback from the analysts will be used to establish GUI development priorities.