To: IVS-WG4
From: John Gipson
Date: 21-October-2008
Re: VLBI data and Meta-Data Storage and Use

# 1. Introduction

Some of the goals of WG-IV are:

1. Completeness. We want to archive all VLBI data so that, if necessary, it can be reanalyzed or reprocessed as techniques and our knowledge improves.
2. Reduce redundancy. VLBI databases have tremendous redundancy. Reducing this would make storage and transmission easier.
3. Data availability at different levels of abstraction.
4. Ability to access only subsets of the data which are most relevant.
5. Flexibility—ability to add new data types or new models.

I address these issues in this note and propose a way meeting these goals. The proposed scheme is sufficiently flexible that it can be used for existing VLBI data as well as the needs of VLBI2010 and the foreseeable future.

As a preliminary to addressing these different, and to some extent competing goals, it is useful to characterize VLBI data and meta-data using different characteristics. I mean "data" in broad sense to include both measured data, and physical and geophysical effects and models, such as nutation/precession, source maps, and thermal deformation.

# 2. Data Characteristics

The following several tables summarize different characteristics of the VLBI data. The particular characteristics I examine are arbitrary. The sole justification is usefulness. There may well be other more useful ways of looking at the data. I welcome feedback on other ways of characterizing the data, or if you think I have left out something important.

Each table has the name of the characteristic followed by a brief description. For example, the first characteristic is **Scope**: *Breadth of Data.* Rows in the table consist of a particular value for the characteristic, followed by an explanation and then several examples. The values usually go from more general to more specific. For example, for **Scope** these go from Universe (applicable to the universe) to Observation, which is only applicable to a given observation. Most examples are drawn from information in the current VLBI databases. A few examples involve data which is not in the VLBI database, but perhaps should be available. Each table is followed by some further considerations.

**Table 1. Scope**
Breadth of data

| Name | Comments | Examples |
|---|---|---|
| Universe | True for a broad section of the universe. | Speed of light<br>Pi |
| Global | True for all VLBI data. | Domes # |

| | | |
|---|---|---|
| Session | Applicable for a single session.<br>Type 1 lcodes. | Principle investigator.<br>Name: R1345<br>Frequency Sequence<br>Correlator<br>A priori station positions |
| Scan | Applicable for a given scan | Source<br>Nominal Start Time<br>Rotation parameters<br>Loading corrections<br>Position of the planets |
| Scan-Station | True for all observations involving a given station in a given scan. | Pointing<br>Loading corrections<br>Cable Cal<br>Phase Cal<br>Mapping functions |
| Observation | Applicable for a given observation | Observed delay<br># of ambiguities<br>Editing criteria |

**Considerations/Comments.**

1. There is no need to store truly universal items.
2. This description is very similar to Anne-Marie Gontier's classification under Pivex, where there are Session, Scan, Station and Baseline kinds of data.
3. In the current VLBI databases, Universal, Global and Session scopes all are Type 1 lcodes.
4. Scan, Scan-Station and Observation Scopes are all Type 2 lcodes[1].
5. As we descend in the hierarchy the cost of storing extra data increases. For example, if we have something that is seldom used but occasionally useful, storage cost is a minimal consideration if the scope is Global or Session. However, if the extra data is Observation dependent, the storage cost grows like number of observations.
6. If you can promote a data item from a lower scope to a higher, you reduce redundancy. For example, in the current VLBI databases, everything is either Session or Observation dependent. Antenna pointing information is stored for every observation. This means that in an N-station scan, the elevation of a given station is stored (N-1) times—once for each baseline the station participates. It really needs to be only stored once per scan.

| **Table 2. Time Dependence**<br>What is the time-tag of the data? | | |
|---|---|---|
| Name | Comments | Examples |
| Session | Doesn't change for the session | Station a priori.<br>Source a priori |
| Scan | Depends only on the scan epoch | Pressure loading.<br>Cable cal |
| Observation | Depends on the observation | |

[1] Type 3 lcodes are really just another kind of Type 2 lcodes. These were introduced when the number of Type 2 lcodes increased above 128.

| | | |
|---|---|---|
| Arbitrary | Have values at other than scan epochs. Must be interpolated/extrapolated to the epoch of observations to be used. | Met data<br>EOP |

**Considerations/Comments.**

1. This is very closely related to Scope.
2. Currently all data (and meta-data) appearing in the VLBI database is either constant for the session, or tied to the epoch of a VLBI observation.
3. The time dependence of the data has important consequences on how the data is stored.
    a. If it is session dependent, it only needs to be stored once per session.
    b. If it is scan dependent, it needs to be stored once per scan.
4. Some data used in VLBI analysis, for example, pressure data, are tied to the time-tags in the database, which in turn are tied to the epochs of the observations. However, the original data is independent of the VLBI observations.
5. It may be useful to have data with time tags before (and after?) the session. For example, for thermal deformation corrections there is evidence that there is a lag of ~2 hours between changes in temperature and antenna deformation.

**Table 3. Primitiveness**

Is the data derivable from other factors?

| Name | Comments | Examples |
|---|---|---|
| Constants | Mathematical and Physical constants. | Pi<br>Speed of light. |
| Correlator Output | Never changes unless the data is re-correlated. Cannot be predicted in advance. | Correlator output |
| VLBI observables | Results of fringe-fitting correlator output. | Group delay<br>Phase delay<br>SNR<br>Quality Code |
| Editing/Ambiguity | Editing and ambiguity resolution | Editing flags<br># of ambiguities |
| Primitive auxiliary data | Used in analysis but may need further calibration. | Met data. |
| Calibrated auxiliary data | Calibrated version of the above. | Met data<br>Cable cal |
| Predictable | These depend only on the epoch and the stations, and can be calculated in advance. | A priori nutation<br>Pole tide<br>Relativistic corrections<br>Tidal ocean loading<br>Some mapping functions<br>Nutation |
| Un-predictable | These depend on things that are unknown prior to the session. | EOP<br>Pressure loading<br>Non-tidal ocean loading<br>Some mapping functions |

**Considerations/Comments.**

1. Constants do not need to be stored.
2. There are several types of VLBI delay observables: group delay, phase-delay, single band delay. (Similarly for rate.) Only the group delay is presently routinely used. In addition, each band has its own delay.
    a. It may be useful to store each band separately.
    b. Storing data by bands may not be appropriate or relevant for VLBI2010.
3. Editing/ambiguity resolution is a function of the analysis. Different groups may get different results. It is useful to keep this information separate for comparison purposes.
4. Primitive auxiliary data does not make it into the VLBI database. However, we should retain this data for completeness.
5. Calibrated auxiliary data is currently included in the VLBI database.
6. Predictable data does not need to be included since it can be calculated. However, the IVS may consider making some of these available as a service.
7. Unpredictable data may depend on measurements taking during the session, e.g., hydrostatic zenith delay which depends on local met dat. It may also be completely independent, e.g., atmospheric pressure loading.

**Table 4. Frequency of Use**

How often is the data used?

| Name | Comments | Examples |
|---|---|---|
| Very Rarely | Routinely used seldom or only once in the data processing. | Correlator output<br>Information related to correlator processing. |
| Rarely | May be useful, but not routinely used. | Frequency sequence<br>Phase Delays<br>System Temperature |
| Often | Used in most analysis | X-and S-band group delays.<br>Axis offset corrections |
| Calc/solve | Only used in calc/solve | Database residuals.<br>Message flags |

**Considerations/Comments.**

1. Data used only in calc/solve should be removed.
2. It may be useful to split the data based on how often they are used.
    a. Commonly used data, such as delays, would be in one. Roughly speaking, this corresponds to data on NGS cards with some additional data.
    b. Very rarely or rarely used data, such as correlator output files or the frequency sequence, would be in another set.

**Table 5. Natural Form for Storing Data/Models**

Scalar/Vector/Matrix/Other?

| Name | # dims | Comments | Examples |
|---|---|---|---|
| Rotation | 3x3 | Scan dependent rotation matrix.<br>Rotates the entire network at a given time. | Precession/Nutation<br>EOP: UT1, Polar motion |

| | | | |
|---|---|---|---|
| Station Displacement | 3 | Station-Scan dependent.<br>Changes the position of the antenna.<br>May or may not be predictable in advance. | Ocean loading<br>Pressure loading<br>Hydrological loading. |
| Station dependent Delay change | 1 | Most naturally viewed as directly affecting the delay at a given station. | Cable cal<br>Phase cal<br>Relativistic effects<br>Ionosphere<br>Thermal deformation<br>Gravitational Deformation |
| Baseline dependent delay change | 1 | Varies with observation.<br>Depends on the baseline and orientation. | Source structure |

**Considerations/Comments.**

1. Rotation can also be recast as 3-rotation angles. This reduces redundancy.
2. You can recast loading effects as changes in delay. This has the advantage that you store 1-parameter (the change in delay) instead of 3 (change in station position). Doing so violates the desire to "keep all data", since you lose information in this reduction.
3. Some of these effects can be predicted prior to the session. These can be pre-calculated. Others will not be known until after the session.

# 3. VLBI Data for a Given Session

I propose dividing the data in the current VLBI database into several files organized according to the following criteria:

1. Scope and time-dependence. Within a given type, the time dependence is the same.
2. Frequency of use. Separate infrequently used data from frequently used data.
3. Primitiveness. Primitive data is kept separately from derived data.
4. Separate data with physically distinct origins. This simplifies things if models or data change. All you need to do is replace the appropriate file.
5. Separate data depending on how it affects the observable.
6. Separate data which can pre-calculated.

**History/Provenance.** I assume that there is a mechanism for storing descriptive information about the data within the files themselves. This is true for the data formats I have researched: CDF, netCDF, HDF. This descriptive information would include information about what kind of data it is (loading, calibration, etc.) as well as information about the origin.

**Naming Conventions.** Although these are not discussed in this section, I assume that each file will be named in such way that it includes information about:

1. What kind of file
2. What information is in the file?
3. Origin: which institution is responsible?
4. Version

This information would also be stored within the file. The naming convention needs to be specified as part of this work.

The following table describes the different kinds of files. For some of the file types I give examples of data that would be stored.

| | |
|---|---|
| ***Session Files*** | Information common to a session. |
| Description File | Overall information about the session. Not expected to change. Contains descriptive information (session name, PI) and data used in analysis (station position). |
| Analysis Files | Files that include information that may be used in the analysis. For example, pointers to source maps. |
| Associated Files | Various files associated with the session not routinely used in analysis. Schedule file, correlator summary, etc. |
| | |
| ***Scan Files*** | Information common for a scan. |
| Description and Cross Reference File | Information about source, stations, epoch.<br>Cross reference information into station-scan files. |
| PN_rotation | Precession/Nutation. |
| EOP_rotation | EOP rotation matrix |
| Etc. | |
| ***Station Files*** | Contains time dependent information about a station. There are two file types. Station-Scan and Station-Time. The label indicates the time dependence. |
| **Station-Scan Files** | Each entry is associated with a given scan, and the time tag is the time tag of the scan. |
| Pointing | Pointing information |
| Station Displacement files | |
| Ocean Loading | Tidal Ocean loading: Can be calculated in advance |
| Pressure Loading | Depends on external data |
| Hydrology Loading | Depends on external data |
| Station Dependent Delay Files | |
| Cable Cal | Extracted from log files |
| Phase Cal | Extracted from log files |
| Apriori_Atm_Delay | A priori atmosphere delay. Can be calculated based on met data, or weather models. |
| Gravitational Deformation | Modeled or calculated |
| Other Station Files | |
| Atm_Mapping | Atmosphere mapping function |
| Met_Data | Calibrated met data. |
| T-Sys | May separate by band. |
| Etc. | |
| **Station-Time Files** | Information about stations at a given time.<br>Time dependence is arbitrary and explicit. |
| Antenna_Temperature | Antenna temperature (used in thermal deformation). |
| Raw met data | Met data associated with a station. |
| Etc. | |
| **Observation Files** | Information which is applicable to a single observation. |

| | |
|---|---|
| Description and Cross Reference File | Information about the observation including sources, stations, and epoch.<br>Contains mapping from observations into scans. |
| Observable1 | Observables. Group, Phase, Single band delay, SNR, sigma, etc.<br>May separate by band/type, or may keep together. |
| Observable1_Ambiguity | Ambiguity resolution for the observables. |
| Observable1_Editing | Which observations do we keep/discard |
| Baseline_dependent_delay_effects | For example, source structure. |
| Correlator_Information | Information associated with correlation of data, including pointers to Raw correlator files. This information is seldom used. |
| | |
| **Raw Correlator Output** | These files would be seldom used. One might want to occasionally refringe some of the data. In the data is refringed it would effect only the observation files. |

**Comments/Questions**

1. Splitting the data in this way minimizes redundancy. There is a modest extra cost in bookkeeping.
2. Data that can be calculated (e.g., nutation/precession) does not need to be kept. However, there are reasons for doing so:
    a. The only cost in doing so is storage cost.
    b. Making this information available might ensure that people are actually using the correct models.
    c. Analysts who do not want this information would not need to use it.
3. Except in very rare circumstances, a file will not be over-written. Instead, another file with a different version number will be made.
4. I have not specified how much the data is split apart. This is still open for discussion. Considering only the station-scan files, there are several alternatives.
    a. At one extreme, there is one file per-station per-lcode, i.e., one file for Gilcreek's temperature, another for its pressure, etc. This has the following advantages:
        i. If there is a problem with a file (e.g., temperature reading is bad) you only need to replace a single file.
        ii. It is easy to add new kinds of data.
        iii. It is easy to specify only sub-sets of the data.
        iv. Having a finer granularity makes it easier to change things.
    b. An alternative is to gather all station data of a particular kind together: All of the pointing information would be put in one file; the met information in another, etc.
    c. Another alternative is to gather all data for a particular station in one file.
    d. Lastly, you can gather all of the data for all of the stations in one file. This has the advantage that all of the data is in a single place.

# 4. Organizing VLBI Data

Splitting the VLBI data into many different files raises the issue of organization. There are actually two separate issues:

1. Organizing the data on disks.
2. Organizing the data for transmission and analysis purposes.

## *A) Organizing data on disks.*

The directory structure provides a natural mechanism for organizing the data on disk. The figure below presents one example of doing so where under the stations directory the information is organized first by station and then by type. Alternatively you could organize by type and then by station. The advantage of using the directory structure is that it is easy to add new files, and you can use standard utilities to back-up and restore the data. Also by splitting the data into files users can select only those files that they want.

```
\Experiment_Directory
        \SessionDirectory
                \Session Description File_ver1
                \Session File1_ver1
                \Session File2_ver1
        \ScanDirectory
                \DescriptionCrossReference_ver1
                \Scan_PN_rotation_ver1
                \Scan_PN_rotation_ver2
                \Scan_EOP_rotation_ver1
                ….

        \StationDirectory
                \Station1
                        \LogFiles
                        \Pointing_info_ver1
                        \Metdata_ver1
                        \Metdata_ver2
                        \Cable_cal_ver1
                        \Pressure_loading_model1_ver1
                        \Pressure_loading_model2_ver1
                        ....
                \Station2
                        \Logfiles
                        \Pointing_info_ver1
                        \Metdata_version1
                        …
        \ObservableDirectory
                \Description&CrossReference_ver1
                \Observable_type1_ver1
                \Ambigutiy_type1_ver1
                \Ambigutiy_type1_ver2
                \Editing_type1_ver1
                \Editing_type1_ver2
                \Observable_type2_ver1
```

```
            \Editing_type1_ver1
            \Ambigutiy_type2_ver1
            …
            \CorrelatorInformationFile_ver1
        \Raw Correlator output
            Obs1
            Obs2
            Obs3
            …
            ObsN
```

## *B) Organizing Data for Transmission and Analysis.*

Since the VLBI data is split into many different files, and there may be different versions of a file, we need a way of tie-ing these files together. I propose introducing a special kind of file called a "Wrapper" that contains information about what files to use for a given session. Each session could have several different wrappers, and each wrapper would present a consistent way of looking at the data. As versions of files changes, a new wrapper would be created. Different wrappers correspond to different versions of a VLBI database.

At a minimum, the wrapper files would contain:

1. Pointers to files.
2. Information about how to use the file: what is the scope of the file, what kind of data does it contain, etc.
3. It might also contain other information.

This concept is still a little rough, but here is what a wrapper file might look like.

**R1234_gsfc_V100.wrp** Wrapper name gives information about session and origin, as well as version information.

| | | |
|---|---|---|
| ***History/Description*** | Information about this particular wrapper. | |
| | | |
| **Pointers to Session Files** | | |
| Description File | | |
| Source Map file | | |
| | | |
| **Pointers to Scan Files** | What | Info |
| Description&Cross_Reference File | | |
| Precession/Nutation | Pointer to file or model | 2006Model |
| | | |
| **Pointers to Station Files** | | |
| Station 1 Files | | |
| Pointing Information | Pointing Information File | |

| | | |
|---|---|---|
| Displacement file | Ocean loading model | OceanModelName |
| Displacement file | Pressure loading | Source of Model |
| Displacement file | Hydrology loading | Source of Model |
| … | … | |
| Station Delay File1 | Apriori Atmospheric delay | Source of Information |
| Station Delay File2 | Cable Cal | |
| Station Delay File3 | Phase cal | |
| Station Delay File4 | Gravitational induced delay | Source of information (ie, particular model) |
| Station 2 Files | | |
| Pointing Information | Pointing Information File | |
| Displacement file | Ocean loading model | |
| | | |
| | | |
| **Pointers to Observation Files** | | |
| Observation Description File | | |
| Observable | ObservableFile | |
| Editing | EditingFile | Analysis center |
| Ambiguity | AmbiguityFile | Analysis center |
| Observable_delay | Source_Effects | |
| | | |
| Etc. | | |

The IVS would provide a recommended wrapper file for each session. Analysts could use this wrapper file to select which files to download. Another use might be as input to a program that would generate the analog of NGS or PIVEX files.

Wrapper files serve another purpose in that analysts could have their own private versions. These could be use to study the effects of different loading models, source-structure models, etc.

# 5. More Detailed Description of Files

In this section I give more detailed information about the various files. For most cases I give the LCODE. If this information appears in an NGS card, I indicate where it appears.

## *A) Session Files*

These files contain information about the sessions as a whole.

**Session description file** gives information about the session as a whole. This information is not expected to change. Not all information will be of interest to all users. However, because the data is only stored once, we don't need to be too concerned about space.

### Session Description File Lcodes

| Name | Description | NGS Cards | Comment |
|---|---|---|---|
| EXPCODE | Experiment name..... | | |

| | | | |
|---|---|---|---|
| EXPDESC | Experiment description..... | | |
| EXPSERNO | Experiment Serial Number..... | | |
| PI NAME | Agency/contact_person/PI name... | | |
| CORPLACE | Correlator name..... | | |
| CORRTYPE | Correlator type: MK3/MK4/K4 etc. | | |
| RECMODE | Recoding mode..... | | |
| FOURF CF | Control file name for fourfit... | | |
| FOURF CS | Command string used for fourfit. | | |
| NLAGS | Num of lags used for correlation | | |
| APLENGTH | Length of accumul. period in sec | | |
| EpochFirstObs | Epoch first observation | | New (Replacement) |
| EpochLastObs | Epoch last observation | | New (Replacement) |
| NUM4 OBS | Number of observations (I*4) | | |
| Num4 Scans | Number of Scans (I*4) | | New |
| # SITES | Number of sites. | SITE | |
| SITNAMES | Site names array. | SITE | |
| SITERECS | Site cartesian coords (m). | SITE | |
| CBL SIGN | Signs of cable cal application | | |
| AXISOFFS | Axis offsets (m). | SITE | |
| ECCCOORD | Eccentricity coords. (m) | | |
| SITEZENS | Site zenith path delays (nsec). | | Used in correlation |
| MEANCABL | Mean cable calibration (seconds) | | |
| # STARS | Number of radio sources. | STAR | |
| STRNAMES | Source names array. | STAR | |
| STAR2000 | J2000 source RAs, decs (rd,rd). | STAR | |

**Session Analysis Files** contain other information that applies to the session as a whole but may change. For example, we may have a file that has a list of source-maps to use for the session. We might have different versions of this file which point to different source maps.

**Associated Files**. Schedule, experiment notes, correlator summary etc. This information is currently absent from the VLBI database, but appears on the session web-pages.

## *B) Scan Files.*

These files contain information about the scan as a whole. They correspond to the @Scan Pivex lcodes.

**Description and Cross Reference File.** This file includes information about the scan, and also information about how to access station dependent information related to a scan.

1. Epoch
2. Unique Scan name.
3. Source
4. Which stations participate in the scan?

5. Station-Scan cross reference.

Strictly speaking, the scan-name is not required for analysis since items 1, 3 and 4 uniquely specify the scan. However, having a unique scan name is useful for many reasons: A) It provides a short-hand way of referring to the scan; B) It could be used as an index value or a key .

Items 4 and 5 can be combined in a single table. The rows of the table correspond to a given scan, and the columns to stations. Zero means the station did not participate in the scan. A non-zero entry indicates that a station participated in a scan, and is a pointer to where the data is stored in the station-scan file (defined below).

The following is an example of part of such a file for a session involving 5 stations.

| Scan # | Epoch | Scan Name | Source | Station-Scan Cross Reference | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2010/Jan/01:00:00:00 | 00:00a | 1015+057 | 1 | 1 | 0 | 1 | 0 |
| 2 | 2010/Jan/01:00:00:00 | 00:00b | 0805+410 | 0 | 0 | 1 | 0 | 1 |
| 3 | 2010/Jan/01:00:01:00 | 01:00 | 1823+689 | 2 | 2 | 2 | 0 | 0 |
| 4 | 2010/Jan/01:00:01:40 | 01:40 | 1124-186 | 0 | 3 | 0 | 2 | 2 |

The column labeled “Scan #” is not part of the table, but is inferred.

Stations 1, 2, 4 participate in the first scan which starts at 00:00. For each of these stations, the information associated with this scan is found in the 1st entry of the station-scan table.

Stations 3 and 5 participate in the second scan, which also starts at 00:00.

Stations 2, 4 and 5 participate in the fourth scan. The information for station 2 is in the third entry of the station-scan tables. For stations 4 and 5 it is in the second entry of the appropriate tables.

There are other ways of storing the information about the stations involved in a scan. For example, you could associate a 2xNumScanStat array with each scan, where NumScanStat is the number of stations in the scan. For each double, the first entry indicates the station number involved in the scan, while the second entry corresponds to where the data is stored. In the above example, the array associated with the first scan would be ((1,1), (2,1), (4,1)). The array associated with the 4-th scan would be ((2,3), (4,2), (5,2)). Such a scheme might result in a modest space savings.

Note that this file, together with the Observation Description and Cross Reference File replaces the following lcodes:

## Scan Description File Lcodes

| Name | Description | NGS Cards | Comment |
|---|---|---|---|
| STAR ID | Radio source name..... | CRD1 | |
| UTC TAG | Epoch UTC YMDHM..... | CRD1 | |
| BASELINE | Ref and rem site names..... | CRD1 | |

**Other Scan Files.** These files contain other information associated with the scan as a whole. These files have no explicit time tags. Instead, they use the same time tags as the Scan-

Description file. The first "entry" of each file is associated with the first scan; the second with the second scan, etc. A few examples are presented below.

**Nutation/Precession-Scan** file. This contains the nutation-precession matrix information on a scan-by-scan basis.

**EOP-Scan** file. This contains EOP information on a scan-by-scan basis.

Information for both of these is currently in the database. In fact, the databases have information about several different kinds of nutation. Neither one of these files would need to be provided by the IVS, although the IVS may do so as a service. In addition, the correlators use this information in processing the data, so it might be useful to make this available.

## *C) Station Files*

These files contain much of the information stored in the database. These correspond to the "@Station" lcodes in the PIVEX formalism.

Station files store the bulk of the information currently in a data base.

1. Met data.
2. Loading corrections.
3. System temperature.
4. General relativity corrections.
5. Pointing information.
6. Ionosphere corrections. These are currently stored as observation dependent, but they could be reformulated to be Station dependent.

In short, anything that depends only on the station and the time.

In the current VLBI data structure, all of the station-related information is tied to the epoch of the observation. This suggests that we could store most of the station information without explicitly time tags. I call such files station-scan files.

**Station-Scan files** contain information associated with a station where the time tags are implicit. The first entry in each file corresponds to the first scan the station participates in. The second entry to the second scan, etc. Using our above example, the station- scan file for Station-1 would look like:

**Station-1**

| |
|---|
| Info_Scan1 |
| Info_Scan3 |
| … |

While for Station-2 we would have.

**Station-2**

| |
|---|
| Info_Scan1 |
| Info_Scan3 |
| Info_Scan4 |

This has the advantage that you minimize storage space. It has the disadvantage that you need to find the time externally.

**Station-Time** files are similar to station-scan files in that they store station-related information. They differ in that the information has an arbitrary time tag. Such a file would look like:

| **Epoch** | **Station-1** |
|---|---|
| Time1 | Info-time1 |
| Time2 | Info-time1 |
| Time3 | Info-time1 |
| … | … |

This has the advantage that the time tag is explicit, and the disadvantage that you may store more information than you need.

There are some situations where Station-Time files are the only thing you can use. For example, thermal deformation of antennas.

Another place to use Station-Time files would be to store information about clock breaks.

## *D) Observation Files*

These files contain information unique to an observation. These correspond to the Pivex @Baseline lcodes.

**Observation Description and Cross Reference File.** Since we store much of the information at the scan level, we need a file that maps the observations to the scans. This file must also indicate the stations involved in the observation. At a minimum, it contains:

1. Scan index
2. Station 1 index
3. Station 2 index
4. Offset of start from nominal beginning of epoch. (seconds)
5. End of observation from nominal start of epoch. (seconds)

Using the above example, the first several entries in this file would be:

| Obs # | Scan# | Station 1 | Station 2 | Start | End |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 0 | 30 |
| 2 | 1 | 1 | 4 | 0 | 210 |
| 3 | 1 | 2 | 4 | 10 | 70 |
| 4 | 2 | 3 | 5 | 0 | 30 |
| 5 | 3 | 1 | 2 | 0 | 30 |
| 6 | 3 | 2 | 3 | 5 | 60 |
| 7 | 3 | 1 | 3 | 5 | 70 |

The column labeled Obs# is not part of the table, but is inferred.

**Observable Files**

These files consist of the observables at each epoch. It is useful to split these apart depending on the type of observable (X-band, S-band, K-band...) and the type of delay (Group, Phase, Single-band…). Each file would contain the following information:

1. Delay.

2. Sigma
3. SNR
4. Quality code.

**Ambiguity Observable File**
For each observable file there is a corresponding file that contains the number of ambiguities. This is found in the solution phase of the analysis.

**Editing Observable File**
For each observable file there is a corresponding file that contains flags indicating whether the data should be used or not.

**Correlator Auxiliary Observable Files**
These files contain additional information about processing the data. This includes things like:

1. Name of raw correlator output file.
2. Parameters associated with fringe fitting
3. Fringe Version
4. etc.

The over-arching idea is that given this information we should be able to re-fringe the data and reproduce the observables. Most of this information comes from the Type-3 lcodes. This data would not be widely used.

## *E) Raw Correlator Files*

This consists of raw correlator files. This is what is used to fringe-fit the data. The Correlator Auxiliary Observable File would contain pointers to these files.

## 6. VLBI Data Flow

In this section I discuss how the proposed data format would interact with the VLBI data flow. The general idea is that all data would reside on the IVS data center. People take what they need and post the results of their processing analysis. The data flow is most easily presented in the form of a table:

| Trigger | Actions | Who | Comment |
|---|---|---|---|
| Schedule file posted to IVS | Make all files that depend only on schedule:<br>Session/ DescriptionFile<br>Scan/Description and Cross Reference<br>Scan/N/P rotation file<br>Station/Description and Cross Reference<br>Station/Pointing information<br>Obs/DescriptionandCross Reference<br>Obs/Theoretical Delay<br>Make Wrapper Ver0 | IVS | Done automatically.<br><br>These files can be used by correlator in correlation. |
| Data received at Correlator | Download appropriate files from IVS.<br>Correlate the data.<br>Post observable files to IVS<br>Post raw correlator output to IVS<br>Make Wrapper Ver1 | Correlator | |
| Observable files posted to IVS by correlator | Download appropriate files from IVS.<br>Perform solution.<br>Upload ambiguity and editing files.<br>Make Wrapper Ver2 | Responsible Analysis Center | |
| Log files posted to IVS | Extract Cable/Met data.<br>Post to IVS<br>Make WrapperVer3 | Responsible Analysis Center | |
| Slant Path Delay Computer | Post Apriori atmosphere delay and mapping files to IVS. | Vienna<br>Goddard<br>Others | |
| Pressure loading data available | Generate station loading Files<br>Post to IVS<br>Make WrapperVer4 | IVS | Done automatically. |
| Source maps made from session | Post source maps to IVS | USNO<br>Bordeaux<br>Others | Done erratically. |

**Comments.**

1. Since the information currently in the VLBI databases is split into pieces, the pieces can be created and updated independently.

2. Under the proposed scheme, the correlator would not need to run or maintain calc. The information the correlators need to process the data would be automatically generated and posted on IVS. The correlators would download what they need.
3. This is just a "straw-man" proposal for the data-flow. I expect the details to change.
4. Most of the information would be mirrored. Because of their size, the raw correlator files may appear in only one location.

# 7. Questions/Comments/Next Steps

This section contains some thoughts about things that need to be done. I organize this as an unsorted list, i.e., the things at the top are not necessarily more important.

5. I propose a maximal division of the station dependent information so that there is one file per station per Lcode.
    a. Is this the right way to go?
    b. Alternative 1. Gather all information for a given lcode (met data) in one file.
    c. Alternative 2. Gather all information for a given station in one file.
6. What about the observables.
    a. I assume that these are separated by band. Is this reasonable?
    b. What kind of information should we keep for each in the "frequently used" file and in the "rarely-used" file?
7. We need to finish specifying the lcodes that we want.
8. Do we need to write a utility to generate NGS cards?
    a. At what step in the processing? After the preliminary files are posted to IVS?
    b. Could we leave this to the users, since the format is open source?
9. Do we need a utility to convert to PIVEX?
10. I see a need for the following programs:
    a. Conversion from current databases into the new format. This should be fairly straightforward.
    b. Extract information from log-files: Cable, Phase, Toys, etc.
    c. Utilities to enable the correlator to use pointing and other information.
    d. Conversion of displacement information into internal format.
    e. What else?
11. I presented a straw-man version of the data flow. How should this change?
12. At what stage should the IVS generate the first Wrapper file?
    a. When the schedule is posted?
    b. After the correlation?
    c. Some other time?
13. Currently there are several different time-tags used. This is our opportunity to fix this. What should we use?
14. What naming conventions should we use? At a minimum, all (most?) files associated with a session should contain the session name. Should they also have more information? I believe they should also contain information about the origin and version.
15. What kind of history information do we want to keep? Should we have a separate history file (or groups of files)?
16. What information should be kept internally in each file?

17. This note focuses primarily on a single session. How do we organize data about all of the sessions?
18. Are there problems with this proposal? Where is it weak? Are there show stoppers?
19. No experiment goes according to plan. There are always station problems of one sort or another. Flux models are off for some sources, so that there are non-detections. Should there be a “clean-up operation” after the correlation so that the VLBI data contains references to only data that is actually taken? This would mean re-generating files.