vgosDB Manual

2015-May-xxx

John Gipson
Chair IVS Working Group IV on Data Structures

WARNING: This document is 2 years out of date and is in the process of being revised. The overall concepts should stay the same, but there are some slight differences in the naming of vgosDB variables and some file names.

We hope to finish the revision of this document by July 31, 2015.

Table of Contents

List of Figures

vgosDB Files

Summary

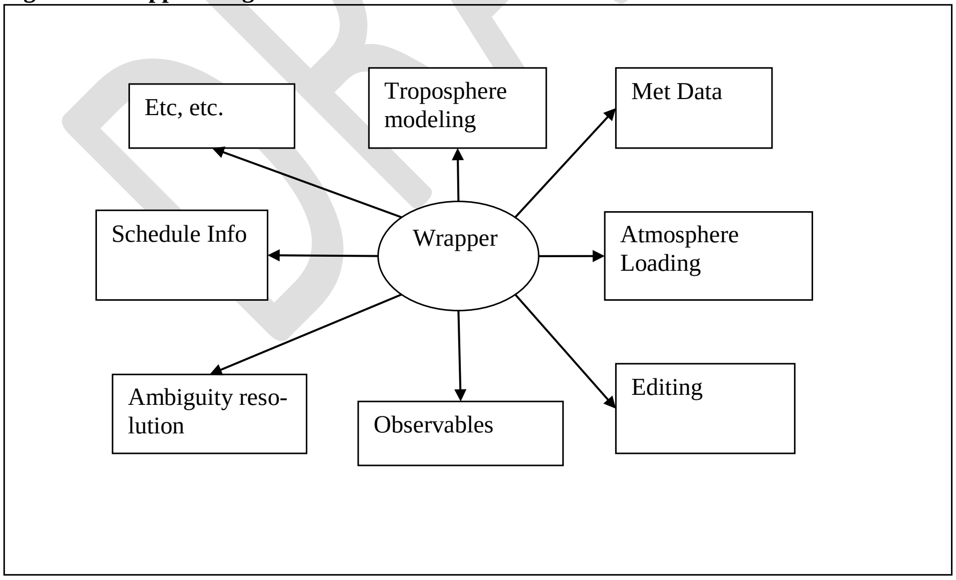
This document presents the vgosDB format to organize and archive VLBI data. VLBI data is organized by session, and stored in open and widely used formats. This scheme has several advantages over storing data in Mark3 databases:

1. **Open**. With a few exceptions, all data is stored in a format that can be handled using software in the public domain. This software has interfaces to many computer languages.
2. **Platform independent**. The data is platform independent, and can be accessed from a variety of operating systems.
3. **Complete**. All data associated with a session is archived.
4. **Separable.** Users can get only the data that they need for analysis. It is possible to replace only a subset of the data, for example, the meteorological data.
5. **Extensible and Flexible**. New data types can be easily incorporated as the need arises.
6. **Compact**. The data is organized by scan to reduce redundancy.
7. **Fast**. Data access is fast.

The general idea is that VLBI data is organized by sessions. Within sessions the data is broken down into files based on the kind of data, the scope of the data, how frequently the data is used, and whether the data may change. The data is stored in either ASCII or NetCDF files.

There is a special kind of file called a **wrapper** which organizes the data within a session. The wrapper contains pointers to the files which contain the actual data. Wrappers allow us to easily expand the kind of data we use. They also allow us to replace data that is obsolete or incorrect.

Figure 1. Wrappers Organize the Data



Definitions

This section gathers in one place definitions of some terms used in this manual. Many of these terms fuller definitions later in this manual.

VLBI Related Names

Observation
Two VLBI stations observing the same source at the same time. There are usually many observations in a scan.

Scan
A set of antennas observing a given source at a common time.

Site
A location with a collection of geoedetic measurement equipment. A site can have none, one, or more VLBI antennas.

Station
A VLBI antenna and associated receiving and recording equipment.

Station Scan
A scan that a particular station participates in.

Mark3 Database related names

The default format for storing and archiving data is the Mark3 database, or MK3-DB.

Mark3 Database
A Mark3 Database is a collection of data related to a single VLBI session that contatians data required to analyze a VLBI session. Data in a Mark3-DB is stored in Lcodes.

Mark3-Lcode
An Lcode is a collection of related data for a single VLBI session. Examples include Station names, Station positions, group delays, SNRs, etc. Each Lcode has associated with it an 8-character name and 3 dimensions. An Lcode can be ASCII, Integer*2, Integer*4, Real*4, Real*8. Lcodes can have upto 3 dimensions.

Mark3 TOC (Table of Contents)
All Mark3 databases contain a Table of Contents indicating what Lcodes they contain.

Mark3 History
All Mark3 databases contain a History summarizing what was done to the database. Ideally this includes what programs were run on the database, the correlator report, what data was added or removed, and anything discovered during the initial analysis of the database.

Lcode-type
There are three types of Lcodes. Type 1 Lcodes contain data that is valid for the entire session. This includes Station Names, Source Names, Station Positions, etc.

Type 2 and Type 3 Lcodes contain that is valid for a single observation. This includes Group Delay, SNR, Quality Code, etc. There is no essential difference between Type 2 and Type 3 Lcodes, and, depending on the particular session, the same data is sometimes stored as Type 2 or a Type 3 Lcode. Type 3 Lcodes were introduced because of peculiarities of the HP operating system when Mark3 databases were first implemented.

Mark3 Database Version
Any time data is added to or removed from a Mark3 database the version number is incremented. Some of the very first databases are at version number 29. The following numbering is the most common one.

1. Version 1 databases are made from the correlator fringe files by the program *dbedit*.
2. Version 2 databases have theoretical and partial values added by the program *calc*.
3. Version 3 databases have cable-cal and met data added by the program *dbcal*.
4. Version 4 database are the results of merging some of the S-band data into the X-band database and doing editing and ambiguity resolution.

Higher level databases result anytime a model is changed. In general when this happens, Lcodes are added to the database so that the Version *N* database contains all of the information (including obsolete information) contained in the Version *N-1* database. It is also possible to remove information from a database.

vgosDB Related Names

vgosDB
A vgosDB is a collection of Binary and ASCII files which contain almost all the information required to process a single VLBI session. Most of the binary files are stored in netCDF format.

Wrapper
A wrapper is an ASCII file that contains pointers to the files in a vgosDB. A wrapper has a name like 12JAN04XA_xxxx.wrp, with 'wrp' indicating it is a wrapper, and 'xxxx' indicating extra characters used to distinguish different wrappers. There can be many wrappers associated with a given vgosDB. Roughly speaking, each wrapper corresponds to a different Mark3-DB version.

History
The history file contains information about one or more processing steps for a vgosDB.

vgosDB Session
A vgosDB session is a VLBI session in which a group of stations with compatible frequency sequencies observe together. This is consistent with the current usage of VLBI session. Most vgosDB sessions are either (roughly) 1-hour or 24-hours.

vgosDB Variable
A vgosDB variable is some distinct VLBI related data which is reasonable to group together. Examples include station names, source positions, observed delay, etc. For convenience we will often not distinguish between the name of the vgosDB variable and the value. There are roughly 400 vgosDB variables. Most Mark3-database lcodes have a corresponding vgosDB variable. Some Mark3 lcodes

vgosDB File
A vgosDB file is a file (usually a netCDF file) that contains one or more vgosDB variables. Most vgosDB files contain a few closely related vgosDB variables. For example, Met.nc contains Temperature, Pressure, Temperature and Humidity at a Station. GroupDelay_bX.nc contains the X-band Group Delay and Sigma.

Scope
Scope is a measure of how broadly applicable some vgosDB variable is. The scope can be Session-, Scan-, Observation- or Station-dependent. Data that does not fit any of these variables is 'Mixed'.

1 Introduction

This document serves many purposes. First and foremost, it describes the vgosDB format. By reading this document you should understand the organization of the vgosDB format. You should be able to access data written in this format. And you should be able to define and write your own types of data.

As much as anything, the vgosDB format is a philosophy for organizing and storing data. Here are some of the guiding principles.

1. VLBI data should be written in an open format.
2. You should know where the data came from.
3. Data should be self describing.
4. Data should be organized by its characteristics.
5. Reducing redundancy is good.
6. Data should be broken up into small enough pieces so that users only need to access that part of the data they want.
7. In most cases, data should never be overwritten.
    a. If you have a better version of the data, for example, a newer geophysical model, then you should create a new version of the appropriate part of the data. This allows you to capture in the future what you have done in the past.
    b. An exception to this is if there is an obvious blunder in the very initial stages of the processing. In this case it make sense to wipe the stage clean and start over. This should *only* be done if there is no danger of someone using the original 'messed-up' data.
8. The number of different files that the data is stored in should be kept reasonably small.
9. Consistency is good.

Sometimes these principles come into conflict and trade-offs had to be made.

To accomplish the above goal, VLBI data is organized by Sessions. These Sessions correspond the current IVS sessions. All of the data for a given VLBI session is stored in a session directory and its sub-directories. Most data is stored in netCDF files. Some data is stored in ASCII files or other proprietary files such as correlator output files. Most of the data files contain only a few closely related data items. This allows users to access only those parts of the data they are interested in. It also means that, in the case of updates, you only need to update a small subset of the data. There are two notable exceptions to this: CorrInfo_xxxxx.nc and CalcInfo_xxxx.nc Each of these files contain many data items used in, respectively, 1) The Correlation process; 2) The calculation of the Theoretical Delay and Rate. For archival purposes it is useful to keep this data. However this data is seldom used, and is not expected to change (very often).

In Section 2 we review the history of IVS Working Group IV.
In Section 3 we summarize how data is currently stored.

2 History of IVS Working Group IV on Data Structures

At the 15 September 2007 IVS Directing Board meeting I proposed establishing a "Working Group on VLBI Data Structures". The thrust of the presentation was that, although the VLBI database system has served us very well these last 30 years, it is time for a new data structure that is more modern, flexible, and extensible. This proposal was unanimously accepted, and the board established IVS Working Group 4. Quoting from the IVS Web site [1]:

"The Working Group will examine the data structure currently used in VLBI data processing and investigate what data structure is likely to be needed in the future. It will design a data structure that meets current and anticipated requirements for individual VLBI sessions including a cataloging, archiving, and distribution system. Further, it will prepare the transition capability through conversion of the current data structure as well as cataloging and archiving softwares to the new system."

2.1 Organization of the Working Group

Table 1. Members of IVS Working Group

Chair	**John Gipson**
Analysis Coordinator	Axel Nothnagel
GSFC/Calc/Solve	David Gordon
JPL/Modest	Chris Jacobs Ojars Sovers
Occam	Oleg Titov Voler Tesmer
TU Vienna	Johannes Böhm
IAA	Sergey Kordobov
Observatorie de Paris/ PIVEX	Anne-Marie Gontier
NICT	Thomas Hobiger Hiroshi Takiguchi

Any change to the VLBI data format affects everyone in the VLBI community. Therefore, it is important that the working group have representatives from a broad cross-section of the IVS community. Table 1 lists the current members of WG4 together with their affiliation or function. The initial membership was arrived at in consultation with the IVS Directing Board. While we wanted to ensure that all points of view were represented we also wanted to make sure that the size did not make WG4 unwieldy. The current composition and size of WG4 is a reasonable compromise between these two goals. My initial request for participation in WG4 was enthusiastic: everyone I contacted agreed to participate with the exception of an individual who declined because of retirement.

2.2 History and Goals

WG4 held its first meeting at the 2008 IVS General Meeting in St. Petersburg, Russsia. This meeting was open to the general IVS community. Roughly 25 scientists attended: ten WG4 members and fifteen others. This meeting was held after a long day of proceedings. The number of participants and the lively discussion that ensued is strong evidence of the interest in this subject.

A set of design goals, displayed in Table 2, emerged from this discussion. In some sense the design goals imply a combination and extension of the current VLBI databases, the information contained on the IVS session Web-pages, and much more information [2].

During the next year the working group communicated via email and telecon and discussed how to meet the goals that emerged from the St. Petersburg meeting. A consensus began to emerge about how to achieve most of these goals.

Table 2. Design Goals of Working Group IV

Goal	Description
Provenance	Users should be able to determine the origin of the data and what was done to it.
Compactness	The data structure should minimize redundancy and the storage format should emphasize compactness.
Speed	Data retrieval should be fsat.
Platform/OS/ Language Support	Data should be accessible by programs written in different languages, running on a variety of computers and operating systems.
Extensible	It should be easy to add new data types.
Open	Data should be accessible without the need of proprietary software.
Decoupled	Different types of data should be separate from each other.
Multiple data levels	Data should be available at different levels of abstraction. For example, levels most users are only interested in the delay and rate observables. Specialists may be interested in correlator output.
Completeness	All VLBI data required to process (and understand) a VLBI session from start to finish should be available: schedule files, email, log-files, correlator output, and final 'database'.
Web Accessible	All data should be available via the Web.

The next face-to-face meeting of WG4 was held at the 2009 European VLBI Meeting in Bordeaux, France. This meeting was also open to the IVS community. At this meeting a proposal was put forward to split the data contained in the current Mark3 databases into smaller files which are organized by a special ASCII file called a wrapper. I summarized some of the characteristics and advantages of this approach. Overall the reaction was positive. In the summer of 2009 we worked on elaborating these ideas, and in July a draft proposal was circulated to Working Group 4 members. The ideas continued to be refined over the next several years.

Because of the desire for the new format to be open, and as a nod to Mark3 database structure, the new format is called vgosDB.

2.3 Overview of New Organization

In this section we present a brief overview of the new format.

2.3.1 Modularization

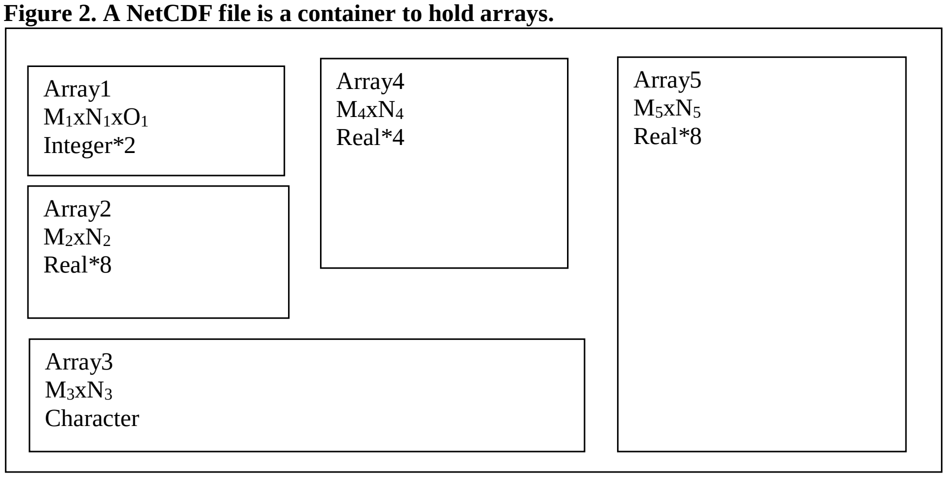
A solution to many of the design goals of Table 3 is to modularize the data, that is to break up the data associated with a session into smaller pieces. These smaller pieces are organized by 'type'; e.g., group delay observable, met-data, editing criteria, station names, and station positions. In many, though not all, cases, each 'type' corresponds to a Mark3 database Lcode.

Different data types are stored in different files, with generally only one or a few closely related data types in each file. For example, it might be convenient to store all of the met-data for a station together in a file. However, there is no compelling reason to store the met-data together with pointing information. Splitting the data in this way has numerous advantages, some of which are outlined below. The first three directly address the design goals. The last was not originally specified, but is a consequences of this design decision.

1. Separable. Users can retrieve only that part of the data in which they are interested.
2. Extensible. As new data types become used, for example, source maps, they can be easily added without having to rewrite the whole scheme. All you need to do is specify a new data type and the file format.
3. Decoupled. Different kinds of data are separated from each other. Observables are separated from models. Data that won't change is separated from data that might change.
4. Partial Data Update. Instead of updating the entire database, as is currently done, you only need to update that part of the data that has changed.

Data will also be organized by 'scope', that is how broadly applicable it is: Does it hold for the entire session, for a particular scan, for a particular scan and station, or for a particular observation. The current Mark3 database is observation oriented: all data required to process a given observation is stored once for each observation. This results in tremendous redundancy for some data. For example, in an N-station scan, there are N*(N-1)/2 observations, and each station participates in N −1 observations. Scan dependent data, such as EOP or source information, is the same for all observations in a scan. However this is stored for each observation, resulting in an N*(N-1)/2 times redundancy. Station dependent data, such as pointing or met-data, which is the same for all observations in a scan, is stored N-1 times instead of once, resulting in an (N − 1)-fold redundancy. Organizing data by scope allows you to reduce redundancy.

Figure 2. A NetCDF file is a container to hold arrays.



2.3.2 NetCDF as Default Storage Format

Working Group 4 reviewed a variety of data storage formats including NetCDF, HCDF, CDF, and FITS. In some sense, all of these formats are equivalent—there exist utilities to convert from one format to another. Ultimately we decided to use NetCDF, because it has a large user community and because several members of the Working Group have experience with using NetCDF. At its most abstract, NetCDF is a means of storing arrays in files. The arrays can be of different sizes and shapes, and contain different kinds of data—strings, integer, real, double, etc. Most VLBI data used in analysis is some kind of array. From this point of view using NetCDF is a natural choice. These files can contain history entries which aid in provenance. Storing data in NetCDF format has the following advantages:

1. Platform/OS/Language Support. NetCDF has interface libraries to all commonly used computer languages running on a variety of platforms and operating systems.
2. Speed. NetCDF is designed to access data fast.
3. Compactness. The data is stored in binary format, and the overhead is low. A NetCDF file is much smaller than an ASCII file storing the same information.
4. Open. NetCDF is an open standard, and software to read/write NetCDF files is freely available.
5. Transportability. NetCDF files use the same internal format regardless of the machine architecture. Access to the files is transparent. For example, the interface libraries take care of automatically converting from big-endian to little-endian.
6. Large User Community. Because of the large user community, there are many tools developed to work with NetCDF files.

2.3.3 Feasibility Demonstration

In August of 2009 John Gipson began a partial implementation of these ideas and wrote software to convert a subset of the data in a Mark3 database into the new format. This subset of data included all of the data available in NGS card format. The subset was chosen because many VLBI analysis packages including Occam, Steelbreeze, and VieVS use NGS cards as input. The GSFC VLBI group made available via anonymous FTP an Intensive, an R1 and a RDV session.

In the fall of 2010, Andrea Pany of the Technical University of Vienna developed an interface to VieVS working with the draft proposal. During this process the definition of a few of the data items needed to be clarified, which emphasizes the importance of working with the data hands on. At NASA's Goddard Space Flight Center, Sergei Bolotin interfaced a variant of this format to Steelbreeze. Steelbreeze uses its own proprietary format, and one motivation for interfacing to the new format was to see if there was a performance penalty associated with using the new format. Bolotin found a small performance penalty of 40 μs/observation.

Meanwhile Gipson continued working on a utility to convert all of the data in all of Mark3 databases into the new format. As there approximately 500 different Lcodes this process took longer than he anticipated. At the same time he began to modify Solve so that it could use the vgosDB format as a replacement for superfiles. The goal was to make the use of vgosDB files as transparent as possible. The software modification process began in the fall of 2011 and by spring of 2012 Gipson had a version of Solve that would work with the vgosDB format with a few restrictions--for example, the "user-partial" feature did not work. I continued to work on solve to remove the restrictions and completed this by the fall of 2012.

There are currently over 5000 24-hour Sessions and 3000 intensive sessions. In the process of converting all of these databases to vgosDB format I uncovered several anomalies and each of these had to be handled in turn. For example, many databases have duplicate Lcodes—the same data is stored in a Type 2 and Type 3 Lcode. In that case the duplicate Lcodes are removed. As another example, all databases contain a Type 1 Lcode NUM OBS which tells how many observations there are in the Type 2 and Type 3 Lcodes. However, several of the older databases actually had fewer observations than was contained in the header, and a few had more.

The process of modifying Solve to use the vgosDB format resulted in a further refinement of the vgosDB specifications.

Timing tests done with Solve and the vgosDB format indicate that there is a performance penalty for smaller databases such as the Intensives and the older databases. We believe that, because of the modular nature of the vgosDB format, there are more I/O operations using the vgosDB format than using superfiles. However, for larger Sessions, such as the RDVs or CONT08 or CONT11, using the vgosDB format is up to 30% faster. In processing large global solutions which include a mix of the overall time required to run a solution using vgosDB or superfiles is about the same.

3 Pre-vgosDB Organization of VLBI Data

Currently the smallest piece of VLBI data that is routinely analyzed is a VLBI session. This information is archived and stored in Mark3 database. These databases contain information used in the analysis of a VLBI session, which are usually 1-hour (intensives) or 24-hours. With very few exceptions, there are usually gaps between sessions, and hence a VLBI session is a natural piece of VLBI data to work with.

3.1 Mark3 Databases

The Mark3 database organizes data by "Lcodes" with each Lcode corresponding to a different data item. The data associated with a given Lcode can be stored as ASCII strings, integer*2, integer*4 or real*8. The Mark3 database was designed to contain all[1] the data necessary to analyze a VLBI session within a single file. The database file contains the observables but it also contains theoretical values, partials and calibrations.

There are two types of Lcodes:

1. Type-1 Lcodes contain data that is applicable for the entire session.
    a. Examples: Station Names, Station Positions, Source Names, Source Position,Session Name, PI, etc.
    b. This information occurs only once in the database.
    c. There are roughly 100 different Lcodes.
2. Type-2 and -3 Lcodes are conceptually identical. Type-3 Lcodes were introduced because of limitations of the HP operating system in the 1980s. These Lcodes contain observation dependent data:
    a. Examples: EOP data, a priori nutation, various partials, delay, rate, sigmas.
    b. The database contains data for each Lcode and each observation, e.g., each observation has an associated EOP value, met values, etc.
    c. There are around 400 different Type-2 and Type-3 Lcodes. A typical Mark3 database will contain around 150 Type-2 and Type-3 Lcodes.

The Mark3 databases are fundamentally organized by observation, as illustrated below.

Table 3. Mark3 databases are organized by session- and observation-dependent data.

Type 1 Lcodes: Session Data						
Source List	Station List	Correlator	Principle Investigator	Flags	Etc...	
Type 2 and 3 Lcodes: Observation Data						
	Lcode1 SourceName	Lcode2 1st Station	Lcode3 2nd Station	Lcode3 EOP	…	LcodeM Observable
Obs1						
Obs2						

[1] Over time this proved impractical, and some of data is now stored in external files. Examples include EOP files, pressure loading, episodic motion, etc.

…						
ObsN						

It is important to note that the Mark3 database format is both a method of organizing data and a means of storing data. Data is organized by Lcodes, where the Lcodes are Session-dependent or Observation dependent. The data is stored in a proprietary format.

The Mark3 database format has some nice features which we do not want to lose. Among these are:

1. Table of Contents. You can easily see what data is available in a given database.
2. Self-descriptive data. Each data item has a brief description of what it is.
3. History. Each database contains a history of its processing.

The specific Lcodes within a database vary depending on the age of the database and how the data was processed. Older databases contain obsolete Lcodes which are relics of how the data was analyzed at the time. In addition databases contain information about the fringing process, and this information is different for each kind of correlator. The number of Lcodes has increased over time as a result of model changes, the desire to use new kinds of data, etc. A consequence of this is that a Mark3 database contains data that is obsolete and never used.

Some problems associated with Mark3 databases are:

1. Requires proprietary software.
2. Only used by the *calc/solve* user community.
3. Redundancy.
   a. Much VLBI data is really scan-dependent, not observation dependent.
   b. There is one database for each band.
4. Mixing of observations and theoretical models.
5. Changing a model or adding new kinds of data means updating the entire database.
6. Difficult or impossible to exchange partial information, i.e., ambiguity resolution or editing criteria.
7. Contains obsolete data and models.
8. Contains data that is very seldom used.
9. Contains data that is *calc/solve* specific.
10. Slow data access which makes it prohibitively time-consuming to use the Mark3 database in large VLBI solutions.

In spite of the above problems, the Mark3 database has been in use for over 35 years which is a testament to the many virtues it has.

3.2 NGS Card Format

Because of the proprietary nature of the Mark3 database an alternative format called "NGS card" format was developed to exchange VLBI information. This consists of a single ASCII file with a series of lines. The top of the file contains header information which describes the session as a whole, such as stations, sources and their positions. This is analogous to the Type-1 Lcodes in the Mark3 database. This is followed by information about the observations. This is analogous to the information contained in the Type-2 Lcodes.

The advantage of the NGS format is that it is fairly easy to write software to parse the file. Some of the disadvantages are:

1. Inflexible. Hard to add new data types.
2. Does not contain all of the VLBI data needed to analyze a solution from the beginning. Hence errors in the initial data editing and ambiguity resolution are 'baked-in' to the data.
3. Machine access is slower than for binary files.

3.3 Other Formats

Because of the speed advantages that storing data in binary files has, most VLBI analysis software uses a custom format specific to the particular software.

For doing large global solutions which combine data from many sessions, *solve* stores data in 'superfiles'. These superfiles are essentially binary dumps of Fortran common blocks which contain subsets of the data in a Mark3 database. The organization is also roughly similar to that of Mark3 databases. One common block contains information common to the session as a whole. Another common block contains information applicable to a given observation. Solve uses this information by reading in the common-block for a particular observation *en-masse*.

Other software packages such as Steelbreeze, Occam and VieVs use their own proprietary format. This makes it difficult to exchange data.

As mentioned above, one of the primary reasons for using a proprietary format is that Mark3 database access is slow. Proprietary binary formats were developed in part as a reaction to this. On the other hand the NetCDF format is designed for fast access. One of the goals of the vgosDB is to encourage the use of a common format for data processing and exchange.

4 Organizing VLBI Data

As mentioned at the start of the Section 3, the smallest piece of VLBI data routinely analyzed is the data contained in a Mark3 database. Each database contains the data for a single session. Sessions are usually 24 hours (standard sessions) or 1 hour (intensives). With a few exceptions such as the CONT series (campaigns where VLBI data is taken over an extended period of time, usually around 2 weeks), there are usually gaps between sessions, and hence a VLBI session is natural piece of VLBI data to work with.

In the most optimistic VLBI2010 scenario, there are never gaps in the observing. Some stations may stop observing for a time (for example, for scheduled maintenance), but there will always be a number of stations observing. This is analogous to the situation in GPS and SLR, where some instruments are always on. However, both of these techniques find it useful to divide data into smaller pieces for analysis.

A crucial difference between VLBI and the other space-geodetic techniques is that VLBI is a co-operative venture—stations must observe in a coordinated manner, i.e., two or more stations must observe the same source at the same time, and the observing modes must be the same or similar. If the observing mode is substantially different, then you cannot correlate the data between two stations, and hence there are no observables.

The generalization of a VLBI session to continuous recording is described below.

4.1 Observing Session

VLBI data is organized by Sessions.

1. An IVS-Session consists of a group of observations from a set of stations observing in a coordinated manner using the same or closely related set of frequencies, bandwidths and sampling characteristics called an "observing mode".
    a. Within a given session, the default is that all stations use the same set of frequencies, bandwidths and sampling.
        i. Some stations may observe only a subset of the common frequencies, or use a different bandwidth. The most common cause for this is equipment failure or lack of equipment.
2. For the VLBI data already taken, each IVS-Session is a separate session.
3. Historically, most sessions are either 24 hours (standard sessions), or 1-2 hours (intensives).
    a. A few of the early VLBI sessions are longer than 24 hours. For example 79AUG03 ran 2 days 14 hours and 56 minutes.
4. As we move to continuous (in time) recording which has no "natural" boundaries, it is useful to artificially partition the data into smaller pieces, such as 1-day or 1-week.
    a. For each session, the considerations of item 1 apply. There is no advantage to mixing data with different observing modes.
5. Sessions are distinguished by the date of the first observation within a session.
    a. For consistency with GPS, we may want to start all sessions at 0-hours UT.
6. Session identification.

a. By default sessions are identified by the date of the first observation, followed by optional characters, e.g., 10JAN04_A.
b. Mark3 databases were identified by the date of the first observation followed by optional characters, e.g., $79AUG03XX, $10JAN04XU. "$" is a special character identifying the file as a database. The penultimate character identifies the band, while the last character identifies the session-type. In data processing, initially there was one database for each band. These are merged together into a single database in a later stage of the processing. In the new scheme all of the bands are contained in a single 'database' from the start.

4.2 Characterizing VLBI Session Data

VLBI Session data consists of all data related to an observing session. One goal of WG-4 is to make all of this data available, and if possible, available online. This would allow re-processing of this data as our knowledge and models improve.

In deciding how to organize the data we need to understand its characteristics.

4.2.1 Useful Ways of Characterizing VLBI Session Data

The following list gives several different ways of characterizing VLBI data.

1. Primitiveness: Is it derivable from other quantities or not?
2. Time dependence:
    a. Is the data time independent? (Station name, DOMES number, etc.)
    b. Is the data effectively constant for the session? (station position)
    c. Is the data time dependent? (Met data, cable cal)
3. Scope: Is it applicable to:
    a. The entire session?
    b. A scan?
    c. A station?
    d. An observation?
4. At what stage in the data processing is the data available? It makes sense to keep together data that is produced at the same time.
    a. The initial processing extracts data from correlator output files and knits them together.
    b. Another step adds calibration and met data.
    c. An independent step calculates the theoretical delay and rate. (b and c occur after a, but can occur in either order.)
    d. Editing and ambiguity resolution are done as a last step.
5. When is the data available?
    a. Available before the session starts? (a priori station positions)
    b. Determined during the session? (met data)
    c. Determined after the session is over? (Source maps)
6. Observables vs calculable.

   a. An observable—that is, depends on or closely related to some measurement? Examples include measurement time-tag, source involved, baselines, observed group delay, etc.
   b. A calculated value independent of the observables. This includes model values like nutation, as well as some partials. (All partials are independent of the VLBI observable, but some depend on other measured quantities, such as the met data.)
7. Use in analysis:
   a. Required—always used.
   b. Convenient—nice to have
   c. Seldom used
   d. Rarely used
   e. Not used in analysis, but may be useful for troubleshooting.
8. Static or Dynamic. This is distinct from time dependence.
   a. Static—once determined these values do not change, ever. Examples include: epoch of the observation, stations and sources involved in the observation, some calculated model values.
   b. Dynamic—values may change over time, or you may use alternatives. Examples include:
      i. Loading corrections
      ii. Ionosphere
      iii. Troposphere calibration.
      iv. Editing.
9. Band. The VLBI observable is characterized by the band.
10. Origin of data:
    a. Internal—measured by the VLBI equipment, or estimated in the analysis. Examples include cable cal, the delays, etc.
    b. External—available from another source.
    c. Either—examples include met data, or ionosphere corrections.
11. Is the data related to other data, either in origin or logically? Some examples:
    a. Usually temperature, pressure and relative humidity come from a single metrological sensor. Because of this it makes sense to group this data together.
    b. Observations and their sigmas are closely related.
    c. There is a wealth of information contained in correlator output files.
12. Required storage format for data: Character, integer, real, etc.

The above considerations suggest that any organizing scheme should have the following characteristics:

1. Separate required data from optional data.
2. Separate commonly used data from less commonly used.
3. Separate static data from dynamic data.
4. Separate theoretical and model values from observables.
5. Organize data by time-dependence and scope.
6. Group data together that is produced at similar stages in processing.
7. Group data together that comes from similar data sources.

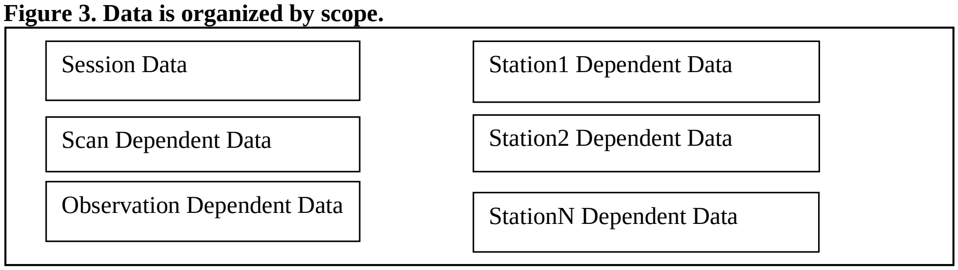
4.3 Organizing Session Data

Here we describe how the above considerations effect how the Session Data is organized.

4.3.1 Organizing Data by Scope

In general, data is organized by scope, that is how broadly applicable the data is. Figure 3, below presents a schematic view of this situation. Data with different Scopes is kept separate from each other.

Figure 3. Data is organized by scope.



The simplest way of doing this is to store the data with different scopes in different directories: Session-dependent data is stored in one directory, Scan-dependent data in another directory, Observation-dependent data in another directory, and additional directories for each station.

4.3.2 Separating Data by Type

Another key feature of data is whether it is an Observable, A priori or Theoretical value. Observables never change. A priori values may change with better information, and Theoretical values may change if the A priori value changes or if we have better models. Because of this it makes sense to separate these different data types.

Theoretical data can be further separated depending on whether it is:

1. Partials derivatives of the delay or rate with respect to some parameter, e.g., EOP or station position.
2. Calibrations. That is models applied to the delay and rate to account for various geophysical and model effects such as EOP, tropospheric delay, etc.
3. Something else--e.g., nutation predictions.

Most theoretical data falls into the first two categories and it makes sense to separate these.

4.3.3 Separating Data by Processing Stage

Another criterion for organizing data is when it becomes available. Data produced at different stages in the processing should be stored in different files. This means that you do not routinely update files. Instead you create a new file that contains different data. This is in contrast to Mark3 databases where a new database is created after every stage of processing:

1. Dbedit takes the correlator output files and knits them together into a Version 1 Mark3 database.

2. Calc reads this database and adds theoretical values including the theoretical delay and rate, and produces a Version 2 database.
3. Dbcal parses the log files and adds cable-cal and met data to the database, producing a Version 3 database.
4. An analyst uses interactive solve to do a preliminary solution. In this solution they resolve ambiguities, do data editing, and determine clock-breaks. This information is added to the database producing a Version 4 database.

At the end of this process you have 4 databases for the same session. Each database contains all of the information contained in the preceding one.

4.3.4 Organizing Observation Dependent Data

There are around 200 different types of observation-dependent data. Most of these are not routinely used in analysis. The vgosDB format groups all of the observation dependent data into related sets and stores them together. For example, the observed Group Delay and its sigma are stored in one file. In organizing this data the following specific principles apply:

1. Conceptually, all data is stored as arrays, and the ordering of the arrays is consistent.
2. The N-th index of all data-items refers to the same observation one.
3. All data-items share a common time-tag--the time-tag of the observation.
4. At a minimum, data is sorted by time-tag and source, with the time-tags being in strictly ascending order. This means that all data for a given scan is contiguous. It is recommended, although not required, that data also be sorted alphabetically by baseline.

To illustrate these concepts consider R1296, the first few scans of which are displayed below.

Table 4. Sked Dump of Beginning of R1296.

```
Source       Start       DURATIONS
name      yyddd-hhmmss   Ft   Hh   Ho   Ny   Tc   Ts   Wf   Wz
0727-115 07274-170000|            40        40   40         |
1611+343 07274-170000| 108   99        92             108   86|
0059+581 07274-170240|                           40   40   40|
1057-797 07274-170319|          105       105               |
0955+476 07274-170416|                40         40   40   40|
1637+574 07274-170556|                           40   40   40|
1334-127 07274-170614| 200                 200               |
2106-413 07274-170723|       99   99                        |
```

The first 16 Observations of this session would be in the following order:

Table 5. First Observations of R1296

Obs #	Source	yyddd-hhmmss	Baseline	
1	0727-115	07274-170000	Ho	Tc
2	0727-115	07274-170000	Ho	Ts
3	0727-115	07274-170000	Tc	Ts
4	1611+343	07274-170000	Ft	Hh

5	1611+343	07274-170000	Ft	Ny
6	1611+343	07274-170000	Ft	Wf
7	1611+343	07274-170000	Ft	Wz
8	1611+343	07274-170000	Hh	Ny
9	1611+343	07274-170000	Hh	Wf
10	1611+343	07274-170000	Hh	Wz
11	1611+343	07274-170000	Ny	Wf
12	1611+343	07274-170000	Ny	Wz
13	1611+343	07274-170000	Wf	Wz
14	0059+581	07274-170240	Ts	Wf
15	0059+581	07274-170240	Ts	Wz
16	0059+581	07274-170240	Wf	Wz
17	1057-797	07274-170319	No	Tc

4.3.5 Organizing Scan Dependent Data

There are around 10 different types of scan-dependent data. The most commonly used one is the Scan Name. Other examples include EOP. In this sub-section, we discuss how this data is organized.

1. Conceptually, all data is stored as arrays, and the ordering of the arrays is consistent.
2. The N-th index of all data-items refers to the same scan.
3. All data-items share a common time-tag--the time-tag of the scan.
4. The order of the scans is the order that the scans appear in the observations.
5. Because of sub-netting, it is possible to have two scans with the same time-tag.

As a concrete example, we again consider R1296.

Table 6. First Few Scans of R1296

Scan #	name	yyddd-hhmmss
1	0727-115	07274-170000
2	1611+343	07274-170000
3	0059+581	07274-170240
4	1057-797	07274-170319
5	0955+476	07274-170416

4.3.6 Organizing Station Dependent Data

There are around 20 different types of station-dependent data. The most commonly used ones are met-data and cable cal. The data for each station is grouped together.

1. Conceptually, all data is stored as arrays, and the ordering of the arrays for a given station is consistent.
2. The N-th index of all data-items refers to the same station-scan.
3. All data-items share a common time-tag.

4. The order of the data is the order that it appears in the observations.
5. In rare cases, because of a bug in the way some correlators compute time-tags, it is possible for a station to have identical two time-tags that involve different scans.
6. In general, the time-tags and scans at different stations are different.

We call the scans that a particular station participates in Station Scans. Table 7 and Table 8 below display the Station-Scans for Hobart and Wettzell. These are the scans that each station participates in, and are labeled in consecutive order. The last column indicates what Session-Scan in the session the particular Station-Scan corresponds to.

Table 7. Station Scans for Hobart in R1296

Station Scan #	name	yyddd-hhmmss	Session Scan #
1	0727-115	07274-170000	1
2	1057-797	07274-170319	4
3	2106-413	07274-170723	8

Table 8. Station Scans for Wettzell in R1296

Station Scan #	name	yyddd-hhmmss	Session Scan #
1	1611+343	07274-170000	2
2	0059+581	07274-170240	3
3	0955+476	07274-170416	5
4	1637+574	07274-170556	6

5 Storing VLBI Data

In the previous sections, we discussed how to organize the VLBI data. In this section, we discuss how to store the data. These issues are somewhat, but not entirely independent. They are not entirely independent, because the storage formats may limit what kind of data can be stored, and the internal organization. For example, NetCDF version 3 and below, does not recognize c-style structures.

In the discussion that follows, we assume that all data that is *routinely* used in VLBI analysis will be stored in either ASCII format or as a NetCDF file.Rarely used data, or data used by only a small subset of analysis centers, may be stored in other formats. Examples include:

1. Correlator output files.
2. FITS files
3. Schedule files, log files, etc.

Because these files are used infrequently in the standard analysis, it is not worth the effort to convert them, together with the software used to access them, into another format.

5.1 General Principles

We list some general principles for handling and creating files that contain VLBI session data.

1. VLBI data files are never (or very, very seldom) overwritten. Instead, a new version of the file is made.
2. VLBI data is grouped depending on use, scope and type of data and whether it is static.
3. Static data is kept separately from non-static data.
   a. Static data never needs to be updated.
   b. Non-static data may need to be.
4. Any program that creates a new VLBI data file indicates:
   a. What the input is.
   b. When the file was created.
   c. Who created it.
5. Data files contain meta-data that:
   a. Gives a brief description of the data.
   b. Indicate the units, where appropriate.

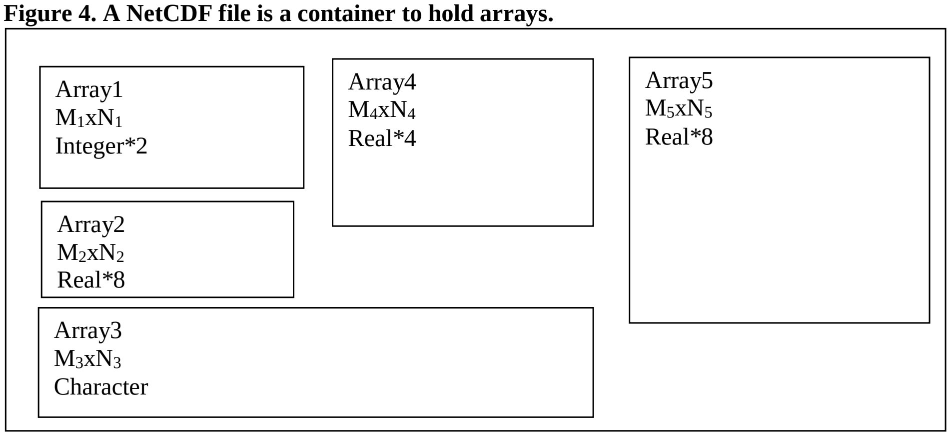
5.2 Some Key Features of NetCDF

We propose that NetCDF be the default format for storing and transmitting VLBI information. This is a data format widely used in the meteorological community. It is open source and has a large user base, interfaces to many languages, and many utilities. It is not a perfect solution though—there are some limitations. Here are some key features:

1. NetCDF is a platform independent storage format with interfaces to most programming languages.
2. NetCDF is designed to store arrays.
3. The valid NetCDF data types are summarized in Table 9.
4. These include most of the variable types used in C and Fortran.

5. Within a given array, all of the data must be of the same type. Corrollary: NetCDF version 3 and below do not know how to handle structures.
6. The NetCDF file extension is ".nc".
7. There is a utility, *ncdump*, which will convert a NetCDF file into a special kind of ASCII file with an extension ".CDL". (CDL= Common data form Description Language.)
8. The utility *ncgen* works in the other direction: it converts an ASCII file written in CDL to a NetCDF file with an".nc" extension.
9. You can store meta-data about the arrays together with the arrays. These includes:
    a. Long-name.
    b. Description
    c. Units
    d. History
10. NetCDF allows users to define arbitrary attributes for data. For example, you can specify the coordinate system as XYZ or UEN, or the time-tag as Implicit. This gives you great freedom in storing meta-data about your data.
11. You can store several different arrays in a single NetCDF file.
    a. The arrays do not need to be of the same type.
    b. The number of dimensions can vary from array to array.
    c. The size of each dimension can vary from array to array.

Figure 4. A NetCDF file is a container to hold arrays.



NetCDF recognizes only a limited number of data types. As mentioned above, all elements in an array must be of the same type. Hence NetCDF does not recognize C-structures or Fortran 90 structures. The types of NetCDF variables are summarized below.

Table 9. NetCDF data types.

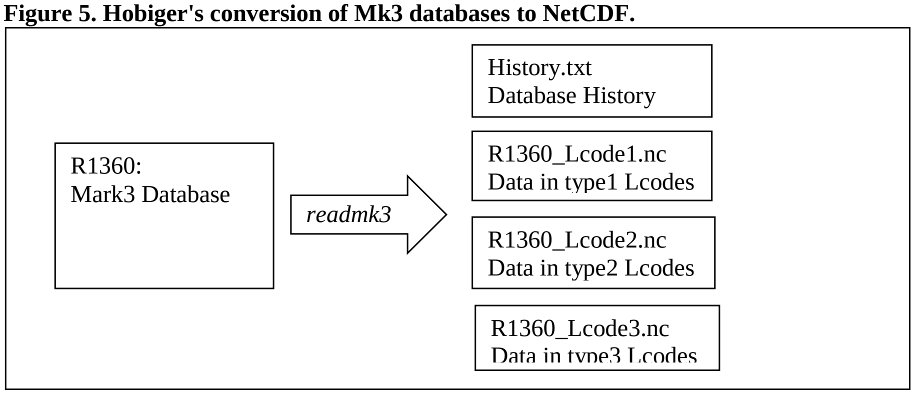
Type	Fortran Type	NetCDF Fortran Mnemonic	# Bits
Byte		NF_BYTE	8
Char	Char	NF_CHAR	8
Short	Integer*2	NF_SHORT	16

Int	Integer*4	NF_INT	32
Float	Real*4	NF_FLOAT	32
Double	Real*8	NF_DOUBLE	64

5.3 Storage Format is Independent of Organization

Thomas Hobiger wrote a routine *readmk3* that converts a Mark3 database into a text file and 3 NetCDF files. The text file contains the history information, and the 3 NetCDF files contain the information contained in the Type1, Type2 and Type3 Lcodes. There is a 1-to-1 correspondence between Mark3 Lcodes and NetCDF arrays. This conversion is illustrated schematically below:

Figure 5. Hobiger's conversion of Mk3 databases to NetCDF.



This mapping preserves the organization of the Mark3 databases. It has the advantage that the data can be easily be accessed using freely available, platform independent software.

It also illustrates that data-organization is distinct issue from data-storage, although they are related.

6 vgosDB Directories

In this section we discuss the organization of vgosDB format data by directories.

6.1 Organization of Session Directories

Table 10. Global Directory Structure

vgosDB/		
	1979/	
	..../	
	2010/	
		10JAN02XK/
		10JAN03XK/
		10JAN04XK/
		10JAN04XA/
		10JAN04XU/
		....
	2011/	

The vgosDB format organizes VLBI data by year and then by session, with each session having its own directory. The session directory name is of the form *YYMMMDDXX* which is similar to the Mark3 database naming convention. Here YY is the two digit year, MMM is the three letter month abbreviation, DD is the two digit day of month and XX is a character designation indicating the particular session. So, for example, the session directory for R1412 which ran on 2010-January-04 is 10JAN04XA. The session directory for the intensive that ran on the same date is 10JAN04XU. A schematic version of this structure is displayed on the right. For reasons of compatibility, the vgosDB directory usually contains the name of the corresponding X-band Mark3 database. However, in contrast to the Mark3 databases that have separate databases for each band, the vgosDB format stores all of the session data for all bands together in the same directory.

6.2 Recommended Session Structure

Within each session directory are several sub-directories which contain a mixture of ASCII and netCDF files. Most of the data proper is stored in netCDF files with an extension ".nc". A few ASCII files are used to give information about or organize the data.

1. Wrapper files organize the data and end with the extension '.wrp'. There may be may many wrapper files for each session, with each wrapper describing a consistent set of the VLBI data. Different software packages may have their wrapper.
2. History files, which end with the extension '.hist', are usually closely paired with wrapper files. These files give information about the processing of the VLBI data. Additional information is also contained within the files themselves.

Figure 1 on the the following page gives an overview of a recommended way to organize the data within within a session. All of the boxes except for Head consist of directories. There are directories called Session, Aux, Observables, etc. There are two exceptions to this. Two exceptions to this rule are the boxes labeled Station and Software. There is one Station directory for each Station, for example, KOKEE, TSUKUBA, WESTFORD, etc. These directories contain station specific data. There is also one Software directory for each VLBI analysis software package, for example, Solve, VieVS, etc.

Note that this is just a *recommended* organization at the current time and may change in the future. The wrapper structure is sufficiently flexible that it can accommodate different structures. If analysis software is written to read the wrapper (as opposed to hard-coding in the location of the directories and the files) it should be able to find the appropriate data wherever it is stored.

Figure 6. Recommended Organization of Session Data

Head.nc
Primitive static data required (or useful) for analysis:
Station Names, Source Names, Number of Observations, Principal Investigator…

/Session
Data used in analysis that is applicable to the session but is not in the Head:
source maps…

/Aux
Data not used in analysis but useful for troubleshooting or other purposes:
Schedule, log files, correlator reports…

/Scan
Scan-dependent data.
Time of Scan, scan name, EOP

/Apriori
Information about stations, sources, etc.
Station Position, Source Position, ….

/Observables
Primitive static, observation data:
Time-tags, Group delay, rate, sigmas

/*Station* (one for each station)
Station dependent data.
One directory for each station.
AzEl, Met, CableCal, Displacement, …

/ObsEdit
Data that depends on editing and ambiguity resolution.
Aambiguity, Full group delay, Editing

/ObsTheoretical
Theoretical delay and rate. Depends on apriori values for stations and sources.
Group delay, rate, sigmas

/ObsCalTheo
Observation dependent calibration data.
Geophysical models

/ObsPart
Observation dependent partials.
EOP, source position, station position

/ObsDerived
Data derived from other data that may be generally useful.
Feedhorn rotation

/*Software* (One for each software)
Directories used for Data of use for some particular software package with one
Solve, Occam, …

/CrossReference
Cross reference information to link Observation-, Scan and Station-dependent data together.

/Correlator
This directory contains correlator fringe files (or the equivalent) and related files.

6.2.1 Head.nc

The file Head.nc contains static information that is applicable to the session as a whole. All of the items are known after processing by the correlator and will not change in subsequent analysis. Most of the data in the Head is available prior to making any observations and is available in the schedule file. Although in principle it might be possible to generate the Head.nc prior to observing, there are some arguments for not doing so. For example, if a station doesn't observe because of equipment problems, there is no need to include it.

We specifically exclude data that is either:

1. Easily calculable, for example, station pointing information;
2. Not static, for example, EOP;
3. That needs to be measured.
4. That can change with better information, such as Station Aprioiri position.

The rationale for the first exclusion is that if it is easily calculable, there is no reason to store it, or at least not to give it priority. The rationale for items 2-4 is that we want the Head to be static-that is, not change in further stages of the analysis.

Most of the data in Head.nc comes from Mark3-DB type-1 Lcodes. It also includes additional data not in the type 1-Lcodes.

Since this data is required for analysis and is not expected to change, it makes sense to gather all of this into one file.

6.2.2 Session

This directory contains information that is applicable to a given session. Examples include:

1. Clock-break information.
2. Source-maps.

6.2.3 Aux

This directory contains data that may be useful for troubleshooting but is not directly used in the analysis of the VLBI data. Examples include schedule files, station logs, correlator reports, preliminary analysis reports, etc. Roughly speaking this is the data that is currently on the IVS session web-pages.

6.2.4 Scan

This directory contains scan-dependent data. This data is applicable to all observations within a given scan. Examples include:

1. Scan names.
2. Scan epochs.
3. UT1 and polar motion.
4. Ephemeris information.

It is useful to split Scan Data into different files depending on the data they contain. Files can be updated independently, and users can choose which kinds of data they want.

6.2.5 Station

There is one station directory for each station, and the name of the directory is identical the IVS name of the station with the exception that spaces are replaced by underscores. Hence “NRAO 85” becomes NRAO_85.

This is data that is time dependent and varies from station to station. Examples of this include:

1. Epochs of the scans that the station participates in.
2. Metrological data.
3. Station dependent calibration, e.g., cable-cal.
4. Pointing data.

The numbers of station dependent items are currently around 10-30. These are grouped in files each of which contains only a few data items. For example, the file Met.nc contains the temperature, pressure and relatively humidity.

There are several kinds of Station Dependent Data:

1. Local sensor data. For example, meteorological data is used in determining the a priori zenith delay, and also some mapping functions.
2. Displacement data. This is measured or modeled displacement of the station given in either XYZ or UEN format.
3. Calibration data that modifies the delay. There are several different kinds:
    a. Internal calibration data, such as cable cal, is measured during a session.
    b. Model results, such as ocean loading, which depend only on the epoch.
    c. External calibration data, such as pressure loading. This is currently not stored in the database, but as an external file.
4. Partials. This is something estimated in the analysis process which is station dependent. Examples include atmosphere dependent partials.

The Mark3 databases include examples of all three kinds of Station Dependent Data.

All current analysis packages require the cable cal data taken from the Mark3 databases. Some analysis packages do not use the partials or the models in the database, and instead compute them as needed.

Note that Station dependent data could be stored as observation dependent data, which is what is done in the Mark3 databases. However, by treating it separately you have the potential to greatly reduce redundancy with the cost of an increase in the amount of book-keeping involved.

In contrast to the data in the Head, it is useful to split Station Data into different files depending on the data they contain. Files can be updated independently, and users can choose which kinds of data they want.

6.2.6 Apriori

This directory contains a priori information needed to process the data. This includes station and source position. It also includes a priori clock offset and rate if known.

vgosDB separates this from the information in Head.nc because occasionally it will change, particularly with new stations or sources. In contrast Head.nc is never expected to change.

This information is also used in calculating the theoretical delay and rate.

6.2.7 Observables

This directory contains items that are directly associated with a given observation and are not expected to change. Examples include:

1. Epoch of observation.
2. Source
3. Baseline
4. Group Delay and sigma.
5. Frequency of observations

The Mark3 database contains around 200 Lcodes that are "Observables".

- Some 'observables' are required by all analysis packages. Examples include the epoch of the observation, the source and the stations involved, and the group delay and sigma.
- Other 'observables' are only used by some analysis packages, such as Group Delay, or only rarely, such as Phase Delay.
- Some 'observables' are used only once and then not needed again. Channel frequencies are used to calculate the effective ionosphere frequency, but are not required thereafter.
- The vast majority of data items are never or very rarely used, but maybe useful for debugging purposes. This includes, for examples, details about the fringing process.

In the case of Mark3 or Mark4 correlators, all of the Observables are originally found in the correlator output files or VEX files. We separate out the most commonly used data into many smaller files, with each file containing only a few data items. For example, the file GroupDelay contains the measure Group Delay and associated sigma. The remaining data items that are not commonly used (but may be useful for debugging purposes) are stored in a single CorrInfo file.

6.2.8 ObsTheoretical

This directory contains the theoretical delay and rate. It may contain other observation dependent or theoretical or model data as well.

6.2.9 ObsEdit

This directory holds data that is the result of initial data editing and ambiguity resolution or that directly depends on this.

1. Number of ambiguities and editing criteria are a result of initial processing.
2. The total Group Delay depends on the number of ambiguities.

Since different groups may use different editing criteria, it is useful to keep this information in separate files.

6.2.10 ObsCalTheo

This directory holds observation dependent data that is used to calibrate the observations. Examples include loading effects, solid earth tide, etc. Many analysis packages compute this ‘on the fly’. However, by pre-computing this information it may be possible to speed up analysis. Furthermore by having the ability to store this information in an external file it is possible to compare the intermediate results of different software packages.

6.2.11 ObsPart

This directory contains observation dependent partials. Examples include, but are not limited to: EOP partials, station-position partials, or source position partials. Many analysis packages compute this ‘on the fly’. However, by pre-computing this information it may be possible to speed up analysis. Furthermore by having the ability to store this information in an external file it is possible to compare the intermediate results of different software packages.

6.2.12 ObsDerived

This directory consists of data items that are of general interest, and that are derived from other data items by simple manipulation of other data items. Examples include the effect of the feed horn rotation on the phase delay.

6.2.13 Software

These directories consist of data items that are specific to some particular software package. Examples include information about how to setup the solutions, what calibrations to apply, and what kind of constraints. There is one directory for each analysis package.

6.2.14 CrossReference

7 This directory contains cross-reference tables that can be the appropriate Scan-dependent and Observation-in this directory can be found in the Section

Linking Observation, Scan & Station Dependent Data.

7.1.1 Correlator

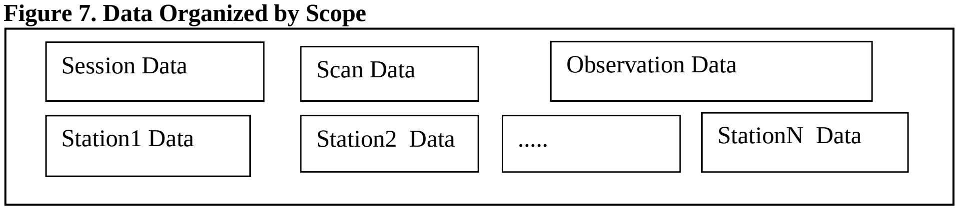
This directory (and subdirectories) contains correlator fringe files and related files or the equivalent. Most of this information is not currently publicly available.

Althous this data is useful for debugging and re-four-fitting, it is not required for anlaysis. Furthermore this data is unavailable for many sessions. Because of this, this data is not required

8 Linking Observation, Scan & Station Dependent Data

In contrast to the Mark3-database where all data is session-dependent (type-1 Lcodes) or observation dependent data (type-2 or type-3 Lcodes), the data in the vgosDB format can be Session, Scan, Station or Observation dependent. Because of this, you need some way of connecting the data items together. We discuss how to do so in this section.

Figure 7. Data Organized by Scope



We start by some general considerations. By assumption, all vgosDB observation data is sorted by time-tag and source (and preferably by baseline as well.) Each observation is specified by:

1. Epoch.
2. Source.
3. Baseline.

Most scans contain many observations, one for each baseline. An N-station scan will contain N*(N-1)/2 observations. We need some way of matching up these observations with the appropriate scan-dependent and station dependent data.

8.1 Using Time-Tags and Sources to Match Data

8.1.1 Associating Observations with Scan Dependent Data

We start by the simple case of examining how to determine which scan a particular observation belongs to. In most cases, knowing the epoch of an observation is sufficient for determining which scan it belongs to. However, for schedules with sub-netting--that is, different antennas can observe different sources at the same time--it may very well happen that you have two scans with the same time-tag. In this case you need to know not only the time-tag of the observation, but the source.

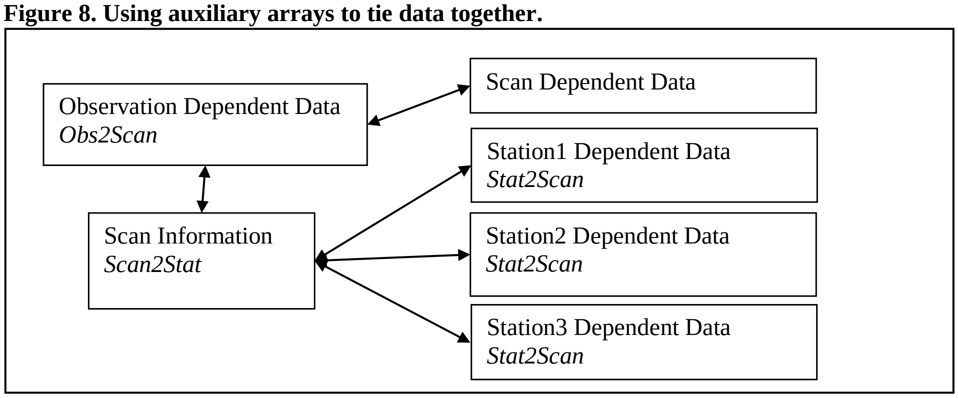
8.1.2 Associating Observations with Station Dependent Data

We now turn to determining how to associate a given station-dependent data-item with a particular observation. In most cases, knowing the time-tag of an observation and a station associated with the observation, it is possible to determine uniquely the associated station-dependent data. However, because of 'bug' in the time-tags of some databases it is possible for a station to participate in two distinct scans with the same time tag. This bug affects about 100 sessions and is discussed in more detail in Appendix A. When this situation occurs, the station's TimeUTC file will have consecutive identical entries corresponding to the two distinct scans. For some station-dependent data such as temperature or pressure which are continuous functions of time, this is not an issue. However, most site-dependent data items depend on the source. Examples of such data items include pointing information, station dependent partials, and cable calibration. In this case you need additional information to determine the appropriate observation. One way of doing so is to match up the sources.

8.2 Tying the Data Together by Cross Reference Tables

As an alternative to matching up observations with scan-dependent and station-dependent data 'on-the-fly' it is also possible to pre-compute cross-reference tables which embody this information. This allows the process of associating some scan or station dependent data with some set of observations to be done much faster. The cross-reference tables *Obs2Scan*, *Scan2Stat* and *Stat2Scan* serve this purpose. There is use is illustrated schematically in Figure 1 below. We want to emphasize that the use of these table is not a requirement to use the vgosDB format.

Figure 8. Using auxiliary arrays to tie data together.



For concreteness in what follows we will use R1296 as a concrete example. The start of this session is presented in Table 11 below.

Table 11. Sked Listing of Start of R1296.

Source name	Start yyddd-hhmmss	DURATIONS Ft	Hh	Ho	Ny	Tc	Ts	Wf	Wz
0727-115	07274-170000			40		40	40		
1611+343	07274-170000	108	99		92			108	86
0059+581	07274-170240						40	40	40
1057-797	07274-170319			105		105			
0955+476	07274-170416				40		40	40	40
1637+574	07274-170556						40	40	40
1334-127	07274-170614	200				200			
2106-413	07274-170723		99	99					

8.2.1 Obs2Scan Array

The Obs2Scan array links observations to the scan they belong in. The number of elements in this array is equal to the number of observations. The value of each element in the array is the scan corresponding to the given observation. Looking atTable 11 Table 11 we note that the first scan involves 3 stations and hence has 3 observations. The second scan has 5 stations and therefore has 10 observations. The third scan has 3 stations and 3 observations, etc. The corresponding Obs2Scan array is simply:

Table 12. Obs2Scan for R1296

Obs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**..**
Scan	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	4	5	..

8.2.2 Obs2Baseline Array

The Obs2Baseline array links observations to the corresponding stations. This is a 2xNumObs dimensional array. Assuming that the observations are sorted by baseline, then the Obs2Baseline array for R1296 looks like the following.

Table 13. Obs2Baseline for R1296

Obs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**..**
Stat1	3	3	5	1	1	1	1	2	2	2	4	4	7	6	6	6	3	4	..
Stat2	5	6	6	2	4	7	8	4	7	8	7	8	8	7	8	9	5	6	

8.2.3 Scan2Stat Array

Scan-dependent data is arranged in order of the time-tags of the scans, and there is one value for each scan. In contrast, for station dependent data we have data values only for the scans that the stations participate in. The array Scan2Stat helps determine which station-dependent data is appropriate for a particular scan. Table 11 shows the Scan2Stat array. The first three columns are not part of the array itself but provide descriptive information. A non-zero value for a station in a particular row indicates that the station participated in that scan. The non-zero entries for each station increase consecutively.

Table 14. Scan2Stat for R1296

			Scan2Stat							
Scan #	Source	TimeTag	Ft	Hh	Ho	Ny	Tc	Ts	Wf	Wz
1	0727-115	07274-170000			1		1	1		
2	1611+343	07274-170000	1	1		1			1	1
3	0059+581	07274-170240						2	2	2
4	1057-797	07274-170319			2		2			
5	0955+476	07274-170416				2		3	3	3
6	1637+574	07274-170556						4	4	4
7	1334-127	07274-170614	2				3			
8	2106-413	07274-170723		2	3					
9	0636+680	07274-170725						5	5	
10	0537-441	07274-171138			4		4			

11	1705+018	07274-171237		3		3			6	5
12	0149+218	07274-171552			5			6		6
13	1144-379	07274-171555	3						7	

For clarity, we have left out the zero values that indicates that a station is not observing. In this example, *Ho*, *Tc*, and *Ts* participate in the first scan. *Ft*, *Hh*, *Ny*, *Wf* and *Wz* participate in the second scan, etc. The first two scans occur simultaneously, illustrating the need to distinguish scans by more than just their time tag.

8.2.4 Stat2Scan Array

The Stat2Scan array helps link station data to the appropriate scan. This array has N-columns corresponding to the N-stations. The number of rows is the maximum number of scans that any station participates in. For a given station, the entries in the columns correspond to the scans that the station participates in. Again using R1296 as an example and just listing the first three rows we have:

Table 15. Stat2Scan for R1296

Station-Scan	Ft	Hh	Ho	Ny	Tc	Ts	Wf	Wz
1	2	2	1	2	1	1	2	2
2	7	8	4	5	4	3	3	3
3	13	11	8	11	7	5	5	5
4	…	…	…	…	…	…	…	…

9 vgosDB Variables and Files

In this section we give more detail about the vgosDB specification. We refer to a particular data item as a vgosDB *variable*. These variables can be any data item associated with a VLBI session such as Correlator, Source Positions, Stations, Observables, etc. vgosDB variables can be ASCII strings, integers, single precession numbers, double precession numbers, or arrays of such items. There are approximately 400 hundred vgosDB variables. Most vgosDB variables are stored in netCDF files. At its most abstract, a netCDF can be viewed as container that stores arrays. Since most VLBI data can be naturally viewed as an array, this is suitable way of storing the data.

Most of the vgosDB files contain a few related variables. For example, all of the met data for a station is stored together. The delay and its sigma are stored together. This approach is in contrast to the Mark3 database or NGS card format where all of the data for a single session is stored in a single file. It has the advantage that:

1. Users have the ability to access only that part of the data they are interested in.
2. You can update part of the data without disturbing the rest.

The main disadvantages are that:

1. You need to impose an external structure on the data. This is done by wrapper files which are discussed separately.
2. It may not be clear what data is available. This disadvantage is overcome by the TOC file which lists all of the data available for a particular session.

9.1 vgosDB Files

A vgosDB file stores a collection of vgosDB variables. Most vgosDB files store only a few closely related vgosDB variables. This allows users to access only those data items that they are interested in.

9.1.1 vgosDB File Naming Conventions.

A vgosDB filename consists of a string of ASCII characters (a-z, A-Z, 0-9, -) separated by underscores. ASCII underscore is a reserved character used to separate fields in the filename. The type of field is determined by the first character after the underscore. The length of each field is arbitrary. The first set of characters before the initial underscore indicates what kind of vgosDB file it is and is called the 'Stub'. Files with the same Stub are 'plug compatible'—that is they contain the same vgosDB variables (perhaps with different values.) Although we anticipate that most of the files will be accessed using software, one of our goals in the file naming convention was to be able to make the file-names human readable and understandable. The table below general form of a vgosDB filename:

Table 16. vgosDB File Naming Convention

Field	Designator (Not case sensitive)	Description
Stub	None	What kind of vgosDB file is this?
Kind	k,K	Kind of data the file contains.
Institution	i,I	Institution or individual
Version	v,V	Version indicator.
Band	b,B	Band of the observation. Required for band-dependent data.

Table 17 below presents some example vgosDB filenames. A complete listing of all of the Stubs is given in Appendix B.

Table 17. Example vgosDB filenames

Filename	Description
10JAN04XA_v004_kSolve.wrp	Wrapper file for session 10JAN04XA for use by Solve analysis package. This is version 4 of the wrapper.
Met.nc	File containing meteorological data.
Cal-Cable.nc	Cable cal data for some site.
Cal-Cable_v2013Jan03.nc	Cable cal data for a site. The version is 2013Jan03.
Cal-Bend.nc	Contribution of general-relativistic bending to the observation.
Dis-OcnLoad.nc	Displacement caused by ocean loading.
Edit_iGSFC.nc	Data editing criteria from GSFC.
Edit_iGSFC_V005.nc	Data editing criteria from GSFC. Version 5.
Edit_iBKG.nc	Data editing criteria from BKG.
Part-Bend.nc	File containing partial with respect to 'bending'.
Part-EOP.nc	Partials with respect to EOP.
Part-HorizonGrad_kNMF.nc	Horizontal gradient at a site using NMF.
SBDelay_bS.nc	Single band delay for S-band.
GroupDelay_bX.nc	Single band delay for X-band.
Freq_bX.nc	File containing frequency information about the X-band data.

9.1.2 vgosDB Header Information

All netCDF vgosDB files contain header information that gives details about when the file was made, who made it, where the data came from, etc. Table 18 lists the optional and required header information. All header information is composed of ASCII strings. The header information serves several purposes. It contains information that serves as a redundancy check to make sure that we are applying the correct data. (For example, the Session-id and the Station serves this purpose.) It provides information about where the data came from. Header information can also identify stray or misplaced files. Table 19 shows the header information from the Head.nc file.

Table 18. vgosDB Header Information

Name	Description	Required	Examples
Stub	Indicates what kind of file this is and the data it contains.	Yes	Head Part Met
CreateTime	When was the file created	Yes	2015/05/27 17:55:59
CreatedBy	Who created it.	Yes	NASA/GSFC
Program	Which program created the information.	Yes	db2vgosDB
DataOrigin	Where did the data come? If present then all data has a common origin. If absent different data items come form different places.	No	R1542wz.log Correlator fringe files
TimeTag	Most time-dependent data uses data contained in an external file TimeUTC. However some data may use time-tags contained within the file.	Yes if data time dependent	Internal StationScan Scan Observation
TimeTagFile	Which file contains time-tag data for this data?	Optional	TimeUTC

	Only required if the time-tag file differs from the default.		
Session	IVS Session name	Optional	XUS79C
Band	Which band is this information for? Only required for band dependent data.	Depends	X S
Station	Which station is this data for? Only required for station dependent data.	Depends	KOKEE WETTZELL
Subroutine	What subroutine created the data.	Optional	
History	An arbitrary ASCII string containing information about how the data was processed.	Optional	

Table 19. Example Header Information

```
Stub = "Head" ;
CreateTime = "2015/05/27 17:55:59" ;
CreatedBy = "John Gipson  NVI, Inc./GSFC" ;
Program = "db2vgosDB            2015May27            John M. Gipson" ;
Subroutine = "Write_Session_File  2015May27 V0.05 JMGipson" ;
DataOrigin = "GSFC database 79AUG03XX version   29" ;
Session = "XUS79C"   ;
```

9.2 vgosDB Variables

There are roughly 500 Mark3 database Lcodes. Each Lcode consists of some VLBI related data item. Some of the Lcodes, such as DEL OBSV (the group delay) contain data that is used by all analysis packages. Many of the Lcodes contain data that is only used by Solve. Many of the Lcodes are obsolete or no longer needed. And some of the Lcodes are only occasionally needed. Removing the obsolete Lcodes still leaves a few hundred data items that must be contained in the vgosDB format. In this section we discuss the general principles for achieving this.

9.2.1 Naming Conventions

A vgosDB variable name can be up to 32 characters long. It must consist of only ASCII upper and lower characters and the numbers 0-9. It cannot include spaces, slashes, periods, dashes, underscores or other special characters. The case of characters is not significant, although it is encouraged that the start of a new word begin with a capital letter for readability: GroupDelay, not groupdelay.

Names are chosen so that they are mnemonic, at least to English language speakers. For example, the vgosDB variable that contains the measured group delay is called GroupDelay. The uncertainty associated with this variable is GroupDelaySig.

vgosDB variables are unique within the following limits:

1. Station dependent variables use the same name at different stations. The temperature data at both Kokee and Wettzell is called TempC.
2. Band-dependent variables use the same names in each band. For example, the observed group delay at both X- and S-band is called GroupDelay. These are distinguished because they are contained in different files.

In another section we give a list of the most common vgosDB variable names. A complete list is given in Appendix C.

9.2.2 Attributes

vgosDB variables have attributes that give additional information about the variables. Some of the attributes are required while others are optional.

Table 20. vgosDB Variable Attributes

Attribute	Description	Type	Required	Examples
Definition	Brief description of the data item.	CHAR*(*)	Yes	Temp in C at local station Measured group delay
LCODE	Corresponding Mark3 Lcode name Although not required, strongly recommended if there is a corresponding LCODE	CHAR*8	Optional	TEMPC DEL OBSV
Units	Appropriate units. Units should be SI	CHAR*(*)	Required for observables	Celsius Second Meter
History	Information about the origin of the data	CHAR*(*)	Optional	
CreateTime	Date the variable was created	CHAR*22	No, but encouraged.	2013/01/26 23:27:28
CalcVer	Version of Calc OR other program used to calculate theoretical values.	CHAR*10	Required for variables calculated by Calc	Calc10.1
REPEAT	Number of time the data repeats. Used to indicate data that is constant over all observations or scans	INTEGER	Optional	4223

As an example, here is a listing of the vgosDB variables contained in the Met.nc.

Table 21. Attributes of Variables in Met.nc

```
 double RelHum(NumScans) ;
   RelHum:LCODE = "REL.HUM." ;
   RelHum:definition = "Rel.Hum. at local WX st (50%=.5)" ;
   RelHum:units = "%" ;
  double TempC(NumScans) ;
   TempC:LCODE = "TEMP C  " ;
   TempC:definition = "Temp in C at local WX station" ;
   TempC:units = "Celsius" ;
  double AtmPres(NumScans) ;
   AtmPres:LCODE = "ATM PRES" ;
   AtmPres:definition = "Pressure in mb at site" ;
   AtmPres:units = "millibar" ;
```

The REPEAT attribute deserves a little explanation. Some variables may have the same value for all scans or observations. Rather than repeat the same value for all the observations, you can specify that the value repeats, and indicate how many times it does so as illustrated in Table 22

Table 22. Use of REPEAT attribute.

```
 double RefFeq(Dim000001) ;
   FreqPhaseReference:LCODE = "REF FREQ" ;
   FreqPhaseReference:REPEAT = 4223 ;
   FreqPhaseReference:definition = "Frequency to which phase is referenced" ;
   FreqPhaseReference:units = "MHz" ;
```

9.2.3 Time Tags

The default time-tags for vgosDB variables is the UTC time. This has the advantage that this is the time-tag used to schedule and correlate the data, and is easily understood by humans. The UTC time is stored as an integer array YMDHM with 5 elements per row corresponding to Year, Month, Day, Hour and a double precision array Seconds.

The time-tags for vgosDB variables can be either implicit or explicit, and internal or external. If the time-tag is explicit then the netCDF file that contains the data also contains the appropriate time-tag. If the time-tag is implicit (which is the case for most vgosDB variables) the time tag is contained in an external file called TimeUTC. There are three distinct types of TimeUTC files depending on the Scope of the variable.

1. The file Scan/TimeUTC.nc contains the time-tag information for all scan-dependent data.
2. The file Observables/TimeUTC.nc contains the time-tag for all observation-dependent data. This time-tag is applicable for ALL observation dependent variables regardless of which directory they are in.
3. Each station directory also contains a TimeUTC.nc file which contains the time-tags for all scans that the station participates in.

9.2.4 Special Treatment of Data With Derivatives

For some vgosDB variables it is useful or required not only to have the variable but also its time derivative (or even its second time derivative). The most common reason for requiring this information is you are using observed Group Rates in your analysis. One possibility would be to define two independent vgosDB variables: one for the variable itself, and a second for the derivative. Instead of doing this, the variable and its time derivative are stored together. For example, the vgosDB variable ElTheo contains both the Elevation and the time derivative of the Elevation. In Fortran this is stored as (NumScan, 2) array, where the second index indicates whether we are dealing with the elevation or its derivatives. The name of the index that differentiates whether it is the function or its derivative is called TimeDim2 or TimeDim3. Most Calibration or Partial files include the time derivative. A few vgosDB variables include the 2nd time derivative

10 Processing Steps and vgosDB Files

In this section we define the common vgosDB files and variables. A complete listing of all of the files is given subsequently. Below we list the steps required in producing a vgosDB session. Thissteps will produce a vgosDB session that can be used in analysis. Apart from making the session and editing the session the order of the steps does not matter: You can calculate the theoretical delays either before or after you add the information from the log files.

At each step in the processing new files are added. At no stage are existing files deleted or modified. If you need to modify a file, you should leave the old file alone and produce a new version of the file. However this should rarely happen for most files.

Table 23. Processing Steps

Step	What Happens
Required Processing Steps	
Making Initial vgosDB Session	Correlator fringe files are read. Head.nc is produced. Observable files are produced. TimeUTC.nc files produced. CorrInfo file is produced.
Adding external information from log file	Cable cal and met data is extracted from log file and put itno files Cal-Cable.nc and Met.nc Other information, e.g., Tsys can also be added at this stage.
Calculation of theoretical delays and rates.	Apriori files produced. Theoretical delay and rates calculated and produced. CalcInfo file produced.
Calculation of Effective Ionosphere Frequency	This uses channel and frequency information contained in CorrInfo.
Ambiguity resolution and editing.	NumAmbig file produced. Total Group delay calculated and GroupDelayFull file produced. X-band Ionosphere calibration calculated and Cal-Ion_bX made.
Optional Processing Steps	
Calculation of partials and calibrations	This can be done at the same time as the calculation of the theoretical delays and rates.
Making cross reference files	This is not required, but is useful for some software packages.

10.1 Required Processing Steps and Files

We start with the required processing and proceed to optional processing steps.

10.1.1 Making the Initial vgosDB Session

This is the stage of processing in which correlator output files are knit together to form the initial set of vgosDB session files. This is equivalent to the current “Version 1” database which contains observational info and little else. All the other processing steps occur after this.

Table 24. Files Made During Initial Processing

File	Description/Comments
Head.nc	Information about the session as a whole that is not expected to change.
The following files are in the Scan Directory.	
TimeUTC.nc	Time-tags for scan dependent information.
ScanName.nc	Names of scan.
The following files are in the Station directory.	
TimeUTC.nc	Time-tags for station dependent data.
Source.nc	Sources that the station is observing. This is only required if it is necessary to distinguish between scans.
The following are in the Observables Directory	
TimeUTC.nc	Time-tags for observation dependent variables
Source.nc	Source for the observation.
Baseline.nc	Baseline for the observation.
The following are band-dependent observables in the Observables Directory. We list just the X-band data. Files for other bands should be present if data is available.	
GroupDelay_bX.nc	Measured group delay and sigma
GroupRate_bX.nc	Measured group rate and sigma
Phase_bX.nc	Measured phase and sigma.
SBDelay_bX.nc	Measured single band delay and sigma
AmbigSize_bX.nc	Ambiguity size
Correlation_bX.nc	Correlation coefficient. Varies between 0 and 1.
QualityCode_bX.nc	Correlator quality code.
SNR_bX.nc	Signal to Noise Ratio
DelayDataFlag_bX.nc	Optional information about the observable. Indicates if the observable is missing or bad.
RefFreq_bX.nc	Reference frequency for phase delay observable.
CorrInfo_kAAA_bX.nc	Miscellaneous information about the observing and fringing process. Most of this information is used only rarely. AAA indicates the kind of correlator: Mark3, Mark4, VLBA, DiFX, etc.

Following the initial stage of processing there are several steps that can occur independently.

10.1.2 Adding External Data by Parsing the Log File

In this stage of processing log files are read in and miscellaneous station-dependent calibrations and assorted information is put into the appropriate vgosDB files. This stage occurs after making the initial session. Currently only two files are created during this step. Other station dependent files such as system temperature may be produced at a later stage.

Table 25. Data Added From Log File

File	Description/Comment
Met.nc	Meteorological information. Required if available.
Cal-Cable.nc	Cable calibration information. Required if available

10.1.3 Calculation of Theoretical Delays and Rates

In this stage we calculate theoretical delay and rate values for the observations. Since these values depend on the a priori station and source position we also generate vgosDB files containing this information. The theoretical delay and rate also depend on the various models used in the calculation. This information is kept track of in CalcInfo.nc. It is useful to keep this information for archival purposes.

Table 26. Adding Required Theoretical Values

File	Description/Comments
The following files are in the Apriori directory	
StationApriori.nc	List of stations and information about the stations.
SourceApriori.nc	List of sources and information about the sources
ClockAprirori.nc	Information about a priori clock offset and rate used in correlation. Only required if this information is available.
The following files are in the Station directory.	
AzEl.nc	Pointing information
FeedRotation.nc	Rotation of the feed horn. Used in phase delay solutions.
The following files are in ObsTheoretical	
DelayTheoretical.nc	Theoretical delay based on the assumed sources, baseline and epoch.
RateTheoretical.nc	Theoretical delay based on the assumed sources, baseline and epoch.
CalcInfo.nc	Information about how 'calc' was setup: what flags were turned on and what models applied.

10.1.4 Calculation of Effective Ionosphere Frequency

This stage calculates the effective ionosphere frequencies. It occurs after *vgosDBMake* but is independent of the *vgosDBca*l and *Calc*.

Table 27. Adding Effective Ionopshere Frequencies

File	Description/Comments
The following files are in ObsDerived	
EffFreq_bX.nc	Effective ionosphere frequencies for group delay, group rate and phase delay.

10.1.5 Editing and Ambiguity Resolution

This stage occurs after all of the previous steps and comprises a series of solutions usually done under manual control. The following things happen during this step.

1. Group Delay Ambiguity resolution. The delay resolution function is periodic, which means that the measured Group Delay is indeterminate. The period of the delay resolution function is called the ambiguity, and is 50-100 ns. The exact value depends on the channel frequencies. As part of this stage of processing we determine how many ambiguities to add to the Group Delay.
2. Editing. We determine which data to keep and throw away.
3. Clock-breaks. We detect the epoch (and perhaps size) of the clock breaks if any.
4. Ionosphere. After a solution is performed for each band we determine the ionosphere correction to be applied at X-band.
5. Phase Delay Ambiguity resolution. This is optional and is rarely done. If the SNR is high enough it may be possible to do a phase delay solution and determine the Phase Delay ambiguities. The size of the Phase Delay ambiguity is the wavelength of the observing band. Hence the measurements must be very-precise in order to do a Phase Delay solution.

Table 28. Data Added After Ambiguity Resoluiton and Editing

File	Description/Comments
The following files are in ObsEdit. Although we only list X-band, S-band is also required.	
Edit_bX.nc	Editing flags for X-band grounp delay.
NumAmbigGroup_bX.nc	Number of Group delay ambiguities.
NumAmbigPhase_bX.nc	Number of Phase delay ambiguities. Only required for a phase delay solution.
GroupDelayFull_bX.nc	Full group delay with ambiguities included
Cal-IonGroup_bX.nc	X-band ionosphere correction for Group Delay.
Cal-IonPhase_bX.nc	X-band ionosphere correction for Phase Delay. Only required if a phase delay solution was done.
The following file(s) are in Session.	
ClockBreak.nc	Information about the epoch (and perhaps size) of real clock breaks.
ClockBreak_bX.nc	Pseudo-clock break for some session.

10.2 Optional Processing Steps

There are a couple of processing steps that are optional. These produce files that are required by *solve* and may be useful for other users, but are not required.

10.2.1 Generation of Cross Reference Tables

One of the advantages of the current Mark3 database structure is that all of the data for a particular observation is grouped together. This has the disadvantage that there is redundancy in the data. In contrast the vgosDB format separates data by Scope. The vgosDB format reduces redundancy, but it also means that you must match up data items. This can be done during processing, but this can also be pre-computed. The cross-reference tables contain the results of this pre-computation. They allow you to easily find the appropriate scan-dependent or station-dependent data for a given observation.

Table 29. Generation of Cross Reference Table

File	Description/Comments
The following files are in the CrossReference Directory.	
ObsCrossRef.nc	Cross reference from observation to scan and to stations.
SourceCrossRef.nc	Cross reference from scan to source.
StationCrossRef.nc	Cross reference from scan to station and from station to scan.

10.2.2 Calculation of Theoretical Models

This stage of processing produces files which are used by the Goddard *solve* analysis packages or contain additional information about the theoretical delays. These files can be grouped into a few specific categories and a catchall category.

1. Partial files start with Part-AAA, where AAA indicates what kind of partial derivative it is. These files contain variables that are the partial derivatives of the delay (and rate) with respect to some parameter of interest. For example, Part-AxisOffset is the Axis-Offset partial derivative. Part-EOP contains the partial derivatives with respect to EOP. Part files can be either station dependent or observation dependent. There can be several different variables in a partial file. The arrays in partial files *always* have two dimensions.

The first index labels the observation (for observation dependent calibrations) or the scan (for station dependent calibrations). The second index labels whether it is a partial derivative with respect to delay or rate.

2. Calibration files start with Cal-AAA, where the AAA indicates what kind of calibration it is. These files contain variables that are calibrations that can be added to or removed from the data. They can be station dependent or observation dependent. As an example, Cal-Cable.nc is the partial file that contains cable cal specific information. A calibration file can contain several different calibrations. The variables in calibration files can have either one or two indices. If they have only one index than the calibration applies only to the delay. If there are two indices then there are calibration values for both the delay and rate.
3. Displacement files start with Dis_kAAA, where AAA indicates what kind of displacement.
4. In addition to the above, there are many other special kinds of files.

Since there are so many different files produced we break the table into smaller pieces.

Table 30. Scan-dependent 'Calc' Files.

File	Description/Comments
The following files are in the Scan directory.	
EOPApriori.nc	A priori EOP for each scan. This is the EOP used in 'calcing' the data.
Ephemeris_kAAA.nc	Contains information about the position of the Sun, Moon, Earth. May contain other planets as well. AAA gives information about the ephemeris.
Nut_kWahr.nc	Calculated Wahr's nutation model.
Nut_k2000PsiEps.nc	IERS 2000 nutation model for Psi and Eps.
Nut_2000XYS.nc	IERS 2000 nutation model for X and Y
Rot_kCF2J2K.nc	Rotation matrix.

Table 31. Station Dependent 'Calc Files'

File	Description/Comments
The following files are in the Station directories	
Cal-AxisOffset.nc	Axis offset calibration.
Cal-NMFDry.nc	NMF mapping function calibration.
Cal-Ocean	Ocean loading calibration.
Dis_kOcean	Displacement caused by ocean loading.
Part-AxisOffset.nc	Axis offset partial derivatives.
Part-NMFDry.nc	Partial derivative with respect to the NMF Dry mapping function

Table 32. Observation Dependent Calibration 'Calc' files.

File	Description/Comments
The following files are in ObsCal and are observation dependent calibrations.	
Cal-Bend.nc	Total calibration due to general relativistic bending.
Cal-BendSun.nc	First order contribution of the sun to bending.
Cal-BendSunHigher.nc	Higher order bending contribution.

Cal-EarthTide.nc	Calibrations due to the earth tide.
Cal-FeedCorrection.nc	Feed rotation. Only effects the rate.
Cal-HFEO_IERS2003.nc	Calibrations due to 2003 IERS HF EOP model.
Cal-OceanLoad.nc	Ocean loading calibrations. This information can be derived from the station-dependent ocean loading calibration.
Cak_kParrallax.nc	Calibration due to parallax
Cal-Poletide.nc	Calibration due to Pole Tide.
Cal-PoleTideOldRestore.nc	If added to PoleTide will restore to an older version of the poletide.
Cal-TiltRemover.nc	Removes contribution of Axis tilt.
Cal-Unphasecal.nc	Calibration to remove the effect of phasecal.
Cal-Whar.nc	Contribution of Wahr nutation model to the delay and rate.
Cal-WobNut.nc	Corrections due to “High frequency nutation”.
Cal-Wobble.nc	Corrections to the delay/rate due to the polar motion.

Table 33. Observation Dependent Partial 'Calc' files.

File	Description/Comments
The following files are in ObsPart and are observation dependent partials.	
Part-Bend.nc	Partial with respect to General Relativistic Bending.
Part-EOP.nc	Partial derivatives for UT1 and Polar Motion.
Part-Gamma.nc	Derivative with respect to gamma.
Part-Nut2KPsiEps.nc	Partial derivatives with respect to nutation Psi and Eps.
Part-Nut2XY.nc	Partial derivatives with respect to X and Y
Part-Parallax.nc	Partial derivative with respect to parallax.
Part-PoleTide.nc	Partial derivative with respect to Poletide.
Part-Precession.nc	Partial derivative with respect to Precession.
Part-RaDec.nc	Partial derivatives for source position.
Part-XYZ.nc	Partial derivative of station positions.

11 Using Wrappers to Organize VLBI Session Data

In contrast to the current Mark3 databases where all (actually most) of the data required to analyze a data is one file, the new scheme proposes dividing the data up into smaller pieces. This allows updating the individual pieces separately and gives great flexibility in what is used. Because we split the data into smaller pieces, there must be another means of organizing the data. Wrappers solve this problem.

11.1 Features of Wrappers

1. A wrapper is a special file which contains
    a. Information about the session
    b. Pointers to files which contain the actual data
    c. It may also contain information about models used in the analysis.
2. A wrapper file is distinguished by the extension ".wrp".
3. The files pointed to by wrappers may reside locally or remotely.
4. A wrapper may contain a pointer to another wrapper file.
5. Wrappers are never over written. Instead, as the results of analysis, or as needs change, a new wrapper is created.
6. For each session the IVS will maintain a wrapper file that contains
    a. Pointers to the data required to analyze the data.
    b. Recommended models used in this analysis.

The IVS is responsible for maintaining the definition of the wrapper file and associated files.

11.1.1 Advantages of the wrappers

1. Flexibility.
    a. Different institutions can use different wrappers.
    b. The format is flexible enough that it could be used by all analysis packages.
2. Extensibility.
    c. New data types can easily be incorporated.
3. Research. By modifying the pointers in the wrapper you can study the effect of using different models.
4. Private wrappers. Individuals can use custom wrappers to study the effect of different models, editing, etc.
5. Easy exchange of information. Breaking the data into smaller pieces and using wrappers to organize the data facilitates the exchange of data. If you are interested in seeing the effect of different editing criteria, all you need to do is get a copy of the editing criteria and modify the wrapper.
6. No obsolete data. The wrapper only needs to point to data that is actually used. As data becomes obsolete, you no longer need to reference it.
7. Ability to reproduce prior analysis. Since nothing is ever overwritten, it should be possible to reproduce the analysis from 5 or 10 years ago.

11.2 Wrapper Format

A session wrapper is an ASCII file that contains pointers to files which contain the actual data.

1. It consists of a series of sections demarcated by "`BEGIN section_name`" and "`END section_name`"
2. If the first character is a "!" or "#" the rest of the line is treated as a comment. Comments can be freely intermixed.
3. Blank lines are ignored.
4. Apart from the `Description` section (see below), all lines in a wrapper consist of

   ```
   Keyword argument(s)
   ```
5. The case of keyword is unimportant. The case of the arguments may or may not be.
6. It is the responsibility of the programs to be able to parse the lines in the wrapper, and take the appropriate action.

11.3 Common Patterns

Most lines in a wrapper consist of a keyword followed by arguments. Most of the time, these keywords specify where certain information can be found. Roughly speaking, each line corresponds to an Lcode, or a collection of Lcodes in Mark3 database.

11.3.1 Pointer to NetCDF array

```
Keyword array_name file_name OptionalArgs
```

`Keyword` specifies what information the array contains, `array_name` is the name of the array within the data file, and `file_name` is the name of the file. For example:

```
StatNames StatNames head.nc
StatXYZ   StatXYZ   head.nc
```

The first line says that the station names array is contained in the array called `StatNames` in the NetCDF file called `head.nc`. The station position is stored in the array called `StatXYZ`, which is also in the NetCDF file called `head.nc`.

11.3.2 Pointer to Several Related Arrays

The above described how to specify an arbitrary array in a NetCDF file. Some NetCDF files contain a series of closely related arrays, and it is useful to specify these all at once

```
Keyword file_name OptionalArgs
```

In this example, `Keyword` specifies the kind of arrays as well as their names, and may specify more than one array. Examples:

```
Head head.nc
…
Met  met.nc
```

The first line specifies that all of the arrays needed for the Head are in the file head.nc. The second line specifies that all the meteorological arrays TempC, AtmPres and RelHum are in the file met.nc. Of course it is still possible to specify one of these arrays individually:

```
TempC TempC met.nc
```

11.3.3 Specifying a Given Model

```
Keyword specific_model model OptionalArgs
```

Here `Keyword` specifies the kind of model, `specific_model` completes the specification, and the literal string `model` indicates that we are specifying a model, as opposed to getting the information from an array. For example:

```
Nutation Wahr model
HF_EOP Gipson96 model
```

Indicates to use the Wahr nutation model , and to use the Gipson1996 HF EOP model.

The advantage of using a similar format to specify both arrays and models is that it is easy to switch back and forth. The line:

```
HF_EOP Gipson96 hfeop.nc
```

Would indicate using the Gipson model, but getting the values from the NetCDF file `hfeop.nc` instead of computing them.

11.4 Wrapper Keywords, File Stubs and Array Names

There is a close relation between wrapper keywords, file stubs and array names. Many array names are the same as wrapper keywords. Filename stubs can be used as wrapper keywords.

11.4.1 Implied Array Name

If the array name is the same as the wrapper keyword, it is permissible to omit the array_name. In this case, if the parsing software detects that the second argument is a NetCDF file, it will assume that the array we are searching for has the same name as the NetCDF keyword. The line

```
TempC TempC met.nc
```

could be replaced by:

```
TempC met.nc
```

Similarly the line

```
AzEl AzEl AzEl.nc
```

could be replaced by:

```
AzEl AzEl.nc
```

11.4.2 Implied Keyword and Array Name

There is a very close connection between filename stubs and wrapper keywords. In fact, all filename stubs are permissible wrapper keywords. Because of this, you can frequently omit both the keyword and array name. This leads to considerable brevity.For example

```
AzEl AzEl AzEl.nc
```

Could be replaced by:

```
AzEl.nc
```

11.5 Changing Default Directory

The default is to store all files associated with a session within the session directory.

1. This default directory can be changed using the keyword `default_dir`:

```
Default_dir dir_name
```

2. The new default directory depends on the first character of `directory_name`:
    a. If the first character of dir_name is a slash ("/"), the new default directory is a top-level directory.
    b. If the first character is not a slash, the new default_directory is a sub-directory of the session directory. If the session directory is "/vgosDB/2010/JAN04XA", and dir_name is "KOKEE", then the new default directory is "/vgosDB/10JAN04XA/KOKEE".
3. Specifying the complete path name for a file always overwrites the default directory.
4. Specifying an external link via ftp overrides default directory.
5. The scope of default_dir is only within a given section of the wrapper file.

In the following the string $SESSION_DIR indicates the session directory.

11.6 Sections of a Wrapper

A wrapper may contain one or more of the following sections:

1. History. Who created the wrapper and when.
2. Description. This describes the wrapper.
3. Session. This contains information about the session, and pointers to data about the session.
4. Scans. This contains pointers to scan-dependent data.
5. Stations. This contains pointers to station-dependent data.
6. Obs. This contains pointers to the observation dependent data.

The organization of a wrapper mimics the overall organization of VLBI session data, and there is a natural correspondence between kinds of VLBI data and sections of the wrapper.

11.7 History

This contains information about who created the wrapper and when, as well as information about the processing of the data. It looks something like:

```
Begin History
!
Begin Program Calc/Solve Processing
Version   Mixed
CreatedBy John M. Gipson
Default_dir History
RunTimeTag 2015/05/07 17:33:51
History    10JAN04XA_kMK3DB_V005.hist
End Program Calc/Solve Processing
!
Begin Program db2vgosDB
Version   2015May03
CreatedBy John M. Gipson
Default_dir History
```

```
RunTimeTag 2015/05/07 17:33:51
History    10JAN04XA_kdb2vgosDB_V20150507173351.hist
End Program db2vgosDB
End History
```

11.8 Description

The description section gives an overview of the content of the wrapper. This section is optional but recommended, Within the description the format is free form.

```
Begin Description
Information about this wrapper.
More information.
….
End Description
```

11.9 Session

The Session Section contains pointers to files that have information about the session as a whole. Each line in the Session Section consists of a keyword followed by optional arguments.

The format of the Session Section is:

```
Begin Session
Session R1412
AltSessionId 0JAN04XA
!
Head.nc
!
Default_Dir Apriori
Eccentricity.nc
Antenna.nc
Station.nc
Source.nc
!
Default_Dir CrossReference
StationCrossRef.nc
SourceCrossRef.nc
!
Default_Dir Session
GroupBLWeights.nc
!
Default_Dir Solve
CalibrationSetup.nc
Misc.nc
CalcInfo.nc
ClockSetup.nc
AtmSetup.nc
ERPSetup.nc
```

```
IonoSetup.nc
CalcERP.nc
SelectionStatus.nc
BaselineClockSetup.nc
End Session
```

The first keyword must be `Session` (R1412 in the above). The first vgosDB file must be Head.nc. This is required because it contains information about the session used elsewhere.

However, since most of this data is required for analysis, there is no need to point to the arrays individually.

11.10 Scan Section

The Scan Section contains information and pointers to files that have information which are scan-dependent. A typical scan section might look like:

```
Begin Scan
!
Default_Dir Scan
TimeUTC.nc
ERPApriori.nc
ScanName.nc
NutationEQX_kIAU2006.nc
NutationEQX_kWahr.nc
NutationNRO_kIAU2006.nc
Ephemeris_kDE405JPL.nc
Rot-CF2J2K.nc
!
Default_Dir Solve
ScanTimeMJD.nc
End Scan
```

The first line changes the default directory to $SESSION_DIR/SCAN. The next line points to the TimeUTC file which contains the time of each scan. The next lines point to other information associated with the scan.

11.11 Station Section

The Station Section contains pointers to files that have station dependent information. There is one Station Section for each station in the session.

A typical Station Section looks like:

```
Begin Station KOKEE
Default_Dir KOKEE
TimeUTC.nc
Cal-Cable.nc
FeedRotation.nc
Met.nc
```

```
Part-AxisOffset.nc
AzEl.nc
Cal-StationOceanLoad.nc
Cal-AxisOffset.nc
Cal-SlantPathTropDry_kNMF.nc
Part-ZenithPathTropWet_kNMF.nc
Part-ZenithPathTropDry_kNMF.nc
Cal-SlantPathTropWet_kNMF.nc
Part-HorizonGrad_kNMF.nc
Dis-OceanLoad.nc
End Station KOKEE
```

Station Dependent files are discussed in more detail below, but it is worthwhile to mention a few.

11.11.1 Station Partial Files

Station Partial files contain the values of partial derivatives of astation-dependent quantities. Many analysis packages compute partials “on-the-fly”, that is, as needed. To speed up the analysis, it is sometimes useful to have the option of pre-computing some or all of these. The Mark3 databases contain several examples of pre-computed partials.

The format for indicating a partial file is:

```
Partial ArrayName FileName.nc
```

If the station is:

1. The first station of the baseline, the partial is multiplied by +1.
2. The second station of a baseline, the partial is multiplied by -1.

The number of entries of ArrayName per epoch are arbitrary, but are the same for all epochs. For the mapping function, we have one entry per epoch. For station position, we have three.

This keyword can appear many times for each Station Section in a wrapper. The only limitation is that the names of the partials is unique.

The `Partial` keyword gives great flexibility in studying the effect of new-partials. Presuming the analysis software is setup to handle it correctly, all the analyst needs to do to study the effect of a new partial is to write a routine to compute it and store it in a NetCDF file.

Unless otherwise over-ridden in the NetCDF file, the default names of the partials are found by concatenating StatName, ArrayName, and array element number, separating them by underscores for clarity. For example, assume we have a partial array “FOO” for KOKEE, and “FOO” has 2 values per epoch. Then the names of the partials would be `KOKEE_FOO_1` and `KOKEE_FOO_2`

11.11.2 Station Calibration Files

Station calibration files contain station dependent data used to correct the observable. This may be the result of:

1. Theoretical calculations, e.g., ocean loading or general relativity.
2. Measurements, for example, cable cal.
3. Externally derived data, for example, pressure loading, or troposphere corrections.

The default unit for all station calibration files is seconds.

If the station is the:

1. First station of the baseline, the calibration is subtracted from the measured delay.
2. Second station of the baseline, the calibration is added to the measured delay.

The format for indicating a calibration file is:

```
Calibration  Array FileName.nc  [Scale]
```

The optional arguments following indicate the time dependence of the data and an optional array of scale values.

The calibration array can have one or more entries per epoch. If the optional Scale is absent, it's elements are assumed to be 1. The calibration applied is the inner-product of the calibration array with the scale array at a given epoch. In Fortran, the calibration applied at epoch j is:

```
Cal(j)=0
Do k=1,DimScale
   Cal(j)=Cal(j)+Array(k,j)*Scale(k)
End do
```

Note that with the above convention, a file could appear as a Partial in one wrapper, and as a Calibration in another. This works well for parameters that are expected to be constant over a session. For example, we could solve for station position in one solution, and then use these values as a calibration in a subsequent solution.

The `Calibration` keyword gives great flexibility in studying the effect of new calibrations. Presuming the analysis software is setup to handle it correctly, all the analyst needs to do to study the effect of a new calibration is write a routine to compute it and store it in a NetCDF file.

11.11.3 Station Displacement Files

Station Displacement Files contain information about station displacement during a session. Examples include ocean loading, pressure loading, and thermal deformation of the antenna. It is the responsibility of the analysis package to handle displacements correctly.

The format for indicating a displacement array is:

```
Displacement Array FileName.nc  [Scale]
```

The optional arguments following indicate the time dependence of the data and an optional scale.

The `Displacement` keyword can appear several times in a Station Section corresponding to different kinds of displacement. The default unit is meters. The default coordinate system is geocentric, unless the array has the attribute `Coordinate` set to `UEN`, in which case the displacement is given in local UEN.

11.12 Observation Section

The Observation Section contains pointers to files that contain information that is observation dependent. This includes data that describe the observation, the observables, the results of analysis, pointers to raw data, etc.

Observation dependent files are organized according to several criteria:

1. Observables are grouped by band.
2. More commonly used data is separated from less commonly used data.
3. The results of analysis are separated from the data.

Some of the wrapper keywords such as GroupDelay and Ambig are modified by adding a band identifier. The observation section of a wrapper looks like:

```
Begin observation
Defalut_DIR OBS
ObsIndex.nc
GroupDelay_bX  Obs_bX.nc
GroupDelay_bS  Obs_bS.nc
Default_Dir ObsMisc
EditGroup_bX   Edit_bX.nc
EditGroup_bS   Edit_bS.nc
GroupAmbig_bX  Ambig_bX.nc
GroupAmbig_bS  Ambig_bS.nc
Iono_bX Iono_bX.
Partial  Par_kEOP.nc

….
End observation
```

Although the number of "columns" in an observation dependent array may vary, the number of rows is always the same, and is the number of observations.

11.12.1 Observation Partial

Observation partial files contain observation dependent data used in the least squares analysis. Examples include things like EOP values.

The general format to include an observation dependent partial file is:

```
Partial ArrayName  FileName.nc
```

An optional argument specifying the time-tag is not since the number of rows is the same as the number of observations.

Commented [DV1]: ????

The presence of this keyword gives you great flexibility in testing new models. This keyword can appear several times for different partials. The only restriction is that ArrayName must be unique for each instance.

11.12.2 Observation Calibration

Observation calibration files contain observation dependent data that is used to "correct" the observable. This may be the result of

1. Theoretical calculations, for example, ocean loading or general relativity.
2. Measurements, for example, cable cal.
3. Externally derived data, for example, pressure loading, or troposphere corrections.

The unit for all station calibration files is seconds.

The general format to include an observation dependent calibration is:

```
Calibration ArrayName FileName.nc [Scale]
```

The presence of this keyword gives you great flexibility in calibrating the data. This keyword can appear several times for different calibrations. The default unit for calibrations is seconds.

11.12.3 Iono File

The Iono.nc file contains ionospheric corrections. There may be one or more files for each band, and not all bands may have a file. These files have names similar to the CommonObs files, e.g., names like `Iono_bX.nc`.

11.12.4 Miscellaneous Observation Files

There are several miscellaneous observation dependent files that are of occasional use. Examples include information about the correlator processing, pointers to the correlator files, etc. These are discussed in further detail below.

Commented [DV2]: Tell exact?

12 Wrapper Dictionary

This section gives a detailed specification of the wrapper grammar. This is organized by section.

12.1.1 Comments and Whitespace

If the first character of a line is “!”, “#” or “/” it is interpreted as a comment. Comments can be freely intermixed.

Consecutive whitespace is (tabs, spaces) is interpreted as a single space. That is, the amount of whitespace between arguments does not matter.

12.2 Common Keywords

12.2.1 Default_Dir

Example:	`Default_Dir Directory_Name`
Description:	Change the default directory.
Comments:	This is valid in all wrapper sections except Description and History.

The default is to store all files associated with a session within the session directory.

1. The new default directory depends on the first character of `directory_name`
2. If the first character:
    a. Is a slash (“/”), the new default directory is a top-level directory.
    b. Is not a slash, the new default_directory is a sub-directory of the session directory
3. Specifying the complete path name for a file always overwrites the default directory.
4. Specifying an external link via ftp overrides default directory.
5. The scope of default_dir is only within a given section of the wrapper file.
6. Default_dir is over-ridden by subsequent instances.

12.3 History Keywords

12.3.1 CreateTimeTag

Example:	`Created 2009Jan04 12:31:01`
Description:	Time-tag when the wrapper was created.
Comments:	This is used only for informational purposes. ***This is required.***

12.3.2 CreatedBy

Example:	`CreatedBy John Gipson`
Description:	Who created the wrapper.
Comments:	This is used only for informational purposes. ***This is required.***

12.4 Session Keywords

12.4.1 SessionName

Example:	`Session 2009Jan04`
Description:	Set the name of the session.
Comments:	If the Session section is present, this keyword is required. The session name is used as a check in all NetCDF files generated specifically for this session. ***The SessionName keyword must be the first keyword in this section.***

12.4.2 Head

Example:	`Head head.nc`
Description:	Point to file that contains the head information.
Comments:	If the Session section is present, this keyword is required.

12.5 Scan Specific Keywords

12.5.1 EOP

Syntax:	`EOP FileName.nc  [Implicit|Explicit]`
Description:	Get the UT1 and polar motion arrays. The optional third argument specifies if the time-tag is implicit or explicit.

Comments:	This is a short-hand for the two lines:
	`PolarMotion PolarMotion eop.nc`
	`UT1         UT1         eop.nc`
	If this line is present, the arrays PolarMotion and UT1 must be present in eop.nc.

12.5.2 Nutation

Syntax:	`Nutation Nutation FileName.nc  [Implicit|Explicit]`
Description:	Extract a nutation array in the new paradigm from a file. The optional fourth argument specifies if the time-tag is implicit or explicit.
Comments:	The example specifies to extract the polar motion from the file eop.nc.

12.5.3 PolarMotion

Example:	`PolarMotion PolarMotion eop.nc  [Implicit|Explicit]`
Description:	Get the polar motion array. The optional fourth argument specifies if the time-tag is implicit or explicit.
Comments:	The example specifies to extract the polar motion from the file eop.nc.

12.5.4 UT1

Example:	`UT1 UT1 eop.nc  [Implicit|Explicit]`
Description:	Get the UT1 array. The optional fourth argument specifies if the time-tag is implicit or explicit.
Comments:	The example specifies to extract the polar motion from the file eop.nc.

12.6 Station Specific Keywords

12.6.1 AtmPres

Example:	`AtmPres AtmPres Met.nc  [Implicit|Explicit]`
Description:	Extract the atmospheric pressure from a file. The optional fourth argument specifies if the time-tag is implicit or explicit.
Comments:	This example extracts the atmospheric pressure from the file Met.nc.

12.6.2 Calibration

Example:	`Calibration CableCal CableCal.nc  [Implicit|Explicit]`
Description:	Extract a calibration array from a file. The optional fourth argument specifies if the time-tag is implicit or explicit.
Comments:	The example extracts the cable calibration.

12.6.3 Displacement

Example:	`Displacement OcnLoad OcnLoad.nc  [Implicit|Explicit]`
Description:	Extract a displacement array. This array contains the displacement of a station due to, for example, loading effects. The optional fourth argument specifies if the time-tag is implicit or explicit.
Comments:	The example extracts the cable calibration.

12.6.4 Met

Example:	`Met Met.nc`
Description:	Extract the met data from a file. The optional fourth argument specifies if the time-tag is implicit or explicit.

Comments:	This keyword indicates that there are three arrays, `AtmPres`, `TempC` and `RelHum` in the file Met.nc. Use of this keyword is a short-hand way of doing:
	`AtmPres AtmPres Met.nc`
	`RelHumid RelHumid Met.nc`
	`TempC TempC Met.nc`

Commented [DV3]: Only two arguments written…. Should it be an optional third argument?

12.6.5 Partial

Example:	`Partial WetMap Atm_kNMF.nc  Implicit`
Description:	Extract an array containing partials from a file. The optional fourth argument specifies if the time-tag is implicit or explicit.
Comments:	This example uses the array WetMap from the file Atm_kNMF.nc as a partial.

12.6.6 RelHumid

Example:	`RelHumid RelHumid Met.nc  [Implicit|Explicit]`
Description:	Extract the relative humidity from a file.
Comments:	This example extracts the relative humidity from the file Met.nc.

12.6.7 TempC

Example:	`TempC TempC Met.nc  [Implicit|Explicit]`

Description:	Extract the temperature from a file.
Comments:	This example extracts the temperature from the file Met.nc

12.7 Observation Specific Keywords

For several of the observation specific keywords you must specify the band. Valid bands are determined by the band-names in head.nc.

12.7.1 ObsIndex

Example:	`ObsIndex ObsIndex.nc`
Description:	Specifies location of ObsIndex File.
Comments:	

12.7.2 Calibration

Example:	`Cal IonGroupCorr Xband_IonoCor.nc`
Description:	Extract a calibration array from a file.
Comments:	

12.7.3 Partial

Example:	`Partial EOP Part-EOP.nc`
Description:	Extract an array containing partials from a file.
Comments:	The above example extracts the array named `EOP` from the file `EOP_Part.nc`.

12.7.4 Ambig

Example:	`Ambig_bX Ambig_bX_v0001.nc`
Description:	Extract ambiguity information for X-band.
Comments:	The above example extracts the ambiguity information for X-band from the file Ambig_bX_v0001.nc

13 Example Wrapper Files

In this Section we present examples of various wrapper files.

13.1 Minimal Wrapper for Intensive Session

Then simplest wrapper is for an intensive, and contains pointers to 4 files. These files contain all of the data required to analyze the session. This wrapper assumes that the analysis package computes the necessary partials.

```
Begin History
Create   2009Jun20-12:22:22
Createdby  JohnGipson
End History
! This is a comment.
Begin Description
This is a simple wrapper file.
It includes the minimum number of pointers required to process a file.
End Description
! -----another comment.
Begin Session
Session I1234
head.nc
End Session
! ***start the station sections.
Begin Station KOKEE
! KOKEE must be one of the station names in Head.nc
Default_dir KOKEE
Calibration Cal-Cable.nc
End Station KOKEE
! WETTZELL must also be specified in Head.nc
Begin Station WETTZELL
Default_dir WETTZELL
! The following illustrates implied keywords based on filename stub.
Cal-Cable.nc
End Station WETTZELL
! **** Start the observation section
Begin Observation
Default_Dir Obs
ObsIndex.nc
Obs_bX.nc
Obs_bS.nc
Default_Dir ObsMisc
Ambig_bX_v0001.nc
Ambig_bS_v0001.nc
Iono_bX.nc
Edit_bX.nc
End Observation
```

13.2 More Extensive Wrapper for Intensive Session

```
Begin History
Create   2009Jun20-12:22:22
Createdby  JohnGipson
End History
```

```
! This is a comment.
Begin Description
This is a simple wrapper file.
It includes the minimum number of pointers required to process a file.
End Description
! -----another comment.
Begin Session
Session I1234
head.nc
End Session
! ***start the station sections.
Begin Station KOKEE
! KOKEE must be one of the station names in Head.nc
Default_dir KOKEE
Calibration Cal-Cable.nc
Atm Atm_kPet.nc
Dis  Dis_kOcnLoad_vStd01.nc
Dis  Dis_kAtmLoad_vStd01.nc
End Station KOKEE
!
! WETTZELL must also be specified in Head.nc
Begin Station WETTZELL
Default_dir WETTZELL
! The following illustrates implied keywords based on filename stub.
Cal-Cable.nc
Atm_kPet.nc
Dis_kOcnLoad_vStd01.nc
Dis_kAtmLoad_vStd01.nc
End Station WETTZELL
!
! **** Start the observation section
Begin Observation
Default_Dir Obs
ObsIndex.nc
Obs_bX.nc
Obs_bS.nc
Default_Dir ObsMisc
Ambig_bX_v0001.nc
Ambig_bS_v0001.nc
Iono_bX.nc
Edit_bX.nc
Part  Part-EOP.nc
End Observation
```

13.3 Minimal Wrapper for R1296 Session

```
Begin History
Create   2009-Jul-14-12:22:22
Createdby  JohnGipson
End History
! -----another comment.
Begin Session
Session R1296
head.nc
End Session
! ***start the station sections.
```

```
Begin Station FORTLEZA
Default_dir FORTLEZA
Calibration Cal-Cable.nc
End Station FORTLEZA
!
Begin Station HOBART
Calibration HOBART/Cal-Cable.nc
End Station HOBART
!
Begin Station HARTRAO
Calibration HARTRAO/Cal-Cable.nc
End Station HARTRAO
!
Begin Station NYALESUND
Calibration NYALESUND/Cal-Cable.nc
End Station FORTLEZA
!
Begin Station TIGO
Calibration TIGO/Cal-Cable.nc
End Station TIGO
!
Begin Station TSUKUBA
Calibration TSUKUBA/Cal-Cable.nc
End Station TSUKUBA
!
Begin Station WESTFORD
Calibration WESTFORD/Cal-Cable.nc
End Station WESTFORD
!
Begin Station WETTZELL
Calibration WETTZELL/Cal-Cable.nc
End Station WETTZELL
! **** Start the observation section
Begin Observation
Default_Dir Obs
ObsIndex.nc
Obs_bX.nc
Obs_bS.nc
Default_Dir ObsMisc
Ambig_bX_v0001.nc
Ambig_bS_v0001.nc
Iono_bX.nc
Edit_bX.nc
End Observation
```

A Buggy Coincident TimeTags

In this section, we describe a ‘bug’ in calculating time tags which results in distinct observations involving the same station being assigned the same time-tags. This bug affects about 100 databases. Table 34 lists part of the schedule R1562 where this bug occurs.

Table 34. Sked Dump of Middle of R1562

```
Source      Start       DURATIONS
name      yyddd-hhmmss   Ft  Hb  Ho  Ke  Kk  Ny  Tc  Ts  Wf  Wz
1759-396 12339-093649|           74  74                         |
3C446    12339-093920|       82      82                  43     |
OJ287    12339-094118|   87              421 113 405         43 421|
2008-159 12339-094125|          117 117                  57     |
0133+476 12339-094427|                       43          43  43 |
2255-282 12339-094516|      242 242  69                         |
```

Prior to early 2013, sessions processed through the Mark4 hardware correlator and the DiFX correlator used different time-tags. The Mark4 correlator used the time-tag of the middle of the period when *all* of the stations were observing. In the above example this is equivalent to taking the start of the scan, and then adding ½ of the shortest observing time. In contrast the DiFX correlator started with the start of the observing time, and then added ½ of the longest scans. The corresponding time-tags for these two cases are displayed below. Depending on the particular scan, sometimes the length of the ½ scan is rounded up or down. For this example, I inferred the rounding by taking the time-tags from the database.

Table 35. Mark4 vs buggy DiFX timetag.

Source	Scan Start	½ Min Scan	½ Max Scan	Mark4 time-tag	DiFX time-tag
1759-396	9:36:49	37	37	9:37:26	9:37:26
3C446	9:39:20	21	41	9:39:41	9:40:01
OJ287	9:41:18	21	211	9:41:39	9:44:49
2008-159	9:41:25	28	58	9:41:53	9:42:23
0113+476	9:44:27	22	22	9:44:49	9:44:49
2255-282	9:45:16	121	35	9:47:17	9:45:51

Note that for the DiFX time-tags, the scans involving OJ287 and 0113+476 have the same time-tag! Since both NyAlesund and Westford participate in both these scans, it looks like both of these sources are observing two different sources at the same time.

For some station dependent variables such as met-data (which varies slowly) this is not that important. For other station dependent variables such as pointing information, mapping functions, or cable-cal, it is important that you know which scan we is the relevant one.

The time-tag bug is actually a bug in the problem DiFX2Mark4. The time-tag bug was discovered by Sergie Bolotin and John Gipson, and the cause of it tracked down by Roger Cappalo who subsequently fixed it.

B vgosDB Data Files Dictionary

In this section we give a description of all of the vgosDB files. In the next section we give a complete list of all of the current vgosDB file types. This is followed by a list of all of the vgosDB files.

B.A Complete List of vgosDB FileTypes

The following is a complete list of the vgosDB file types arranged alphabetically. Each file-type includes a brief description of the data contained into it as well as the Scope. A designation of 'Solve' for the scope indicates that the file is used by Solve, usually in setting up a solution.

The scope of most vgosDB file types is fixed. For example, all AzEl files are Station dependent. The notable examples to this are:

1. 'Cal' files. These files contain calibrations to the data. These can either be observation dependent or station dependent.
2. 'Part' files. These contain partial derivative information. These can either be observation dependent or station dependent.
3. TimeUTC files. These are of three types: Observation-, Scan- and Station-dependent.

Table 36. Dimensions used in vgosDB files.

Dimension Name	Description
NumBaselines	Number of baselines in a session. This is equal to NumStat*(NumStat-1)/2
NumChannels	Number of channels used when observing. Dimension used for some variables in *CorrInfo.nc.*
NumObs	Number of observations made in a session. Dimension used in scopes *Session* and *Obs.*
NumScans	Number of scans made in a session. Dimension used in scope *Session*.
NumSource	Number of sources observed in a session. Dimension used in scope *Session.*
NumStatScan	Station dependent dimension, number of scans. Dimension used in scope *Station*.
NumStation	Number of stations participating in a session. Dimension used in scope *Session*
TimeDim2	Indicates that the vgosDB variable includes a variable and its first time derivative. For example,AzTheo(1,j) is the azimuth of the j-th observation at a station, andAz-Theo(2,j) is the time derivative of the azimuth.
TimeDim3	Indicates that the vgosDB variable includes a variable and its first and second time derivative.

Table 37. Complete List of vgosDB File Types

File	Description	Required?	Dir	Scope
AmbigSize_b?	Size of group delay ambiguties		Observables	Obs
AntennaApriori	Antenna information.		Apriori	Session
AtmSetup	Atmosphere constraints for solution.		Solve	Session
AzEl	Staion Pointing	Y	Station	Station
Baseline	Baseline array	Y	Observables	OBs

BaselineClockSetup	Solution setup.	Y	Solve	Session
Cal-xxxxx	Station dependent Calibration. kXXXX indicates the kind.		Station	Station
Cal-Bend	Observation dependent calibration. kXXXX indicates the kind		ObsCalTheo	Obs
CalcEop	A prioiri EOP used by Calc		Solve	Session
CalcInfo	Information about calc run	Y	ObsTheoretical	Session
CalibrationSetup	Solution setup.	Y (If data available)	Solve	Session
ClockApriori	A priori Clock		Apriori	Session
ClockBreak	Clock Break setup		Session	Session
ClockSetup	Solution setup		Solve	Session
Correlation_b?	Correlation coefficient	Y	Observables	Obs
CorrInfo_b?	Informatation about correlation	Y	Observables	Obs
DelayDataFlag_b?	Is the data good		Observables	Obs
DelayTheoretical	Theoretical Delay	Y	ObsTheoretical	Obs
Dis_kXXXX	Station dependent displacement. kXXXX indicates the kind.		*Station*	Station
EccentricityApriori	Eccentricity Information.		Apriori	Session
Edit	Editing criteria	Y	ObsEdit	Obs
EffFreq_b?	Effective phase frequency	Y	ObsDerived	Obs
EOPApriori	A priori EOP used by Calc		Scan	Scan
EopSetup	Solution setup.		Solve	Session
Ephemeris_kDE405JPL	Ephemeris		Scan	Scan
FeedRotation	Rotation of feed horn		*Station*	Station
FeedRotNet	Net Feed roation. Station1-Staiton2		ObsDerived	Obs
FractC	Fractiional part of a day incluing leapseconds.		Solve	Obs
GroupBLWeights	Additive noise for group delay.		Session	Session
GroupDelay_b?	Measured group delay		Observables	Obs
GroupDelayFull_b?	Fully resolved group delay		ObsEdit	Obs
GroupRate_b?	Measured group rate		Observables	Obs
Head	Information about the session as a whole.	Y	Top Level	Session
IonBits	Ion information used by solve		Solve	Obs
IonSetup	Ionosphere setup in Solve solution		Solve	Session
Met	Meteorological data	Y (if available)	*Station*	Station
Misc	Miscellanous info used by solve.		Solve	Session
NumGroupAmbig_b?	Number group delay ambiguities	Y After doing solution	ObsEdit	Obs
NumPhaseAmbig_b?	Number group delay ambiguities		ObsEdit	Obs
Nut_k2000PsiEps	Nutation information.		Scan	Scan

ObsCrossRef	Links observations to scans		CrossReference	Obs
Part-XXXX	Station dependendent Partial. kXXXX indicates the kind.		*Station*	Station
Part-XXXX	Observation dependent Partial. kXXXX indicates the kind		ObsPart	Obs
Phase_b?	Measured Phase		Observables	Obs
PhaseBLWeights	Noise to add to solution		Solve	Session
PhaseDelayFull_b?	Fully resolved phase delay		ObsEdit	Obs
QualityCode_b?	Correlator quality code	Y	Observables	Obs
RateTheoretical	Theoretical Rate	Y	ObsTheoretical	Obs
RefFreq_b?	Effective frequency for Phase	Y	Observables	Obs
Rot_kCF2J2K	Rottion matrix.		Scan	Scan
SBDelay_b?	Observed SB-delay	Y	Observables	Obs
ScanName	Scan name	Y	Scan	Scan
SelectionStatus	Solution setup.		Solve	Session
SNR_b?	Observed SNR	Y	Observables	Obs
Source	Source observed during scan	Y	Observables	Obs
SourceApriori	A priori source information	Y	Apriori	Session
SourceCrossRef	Links Scans to Sources		CrossReference	Session
StationApriori	A priori station information	Y	Apriori	Session
StationCrossRef	Links stations to scans		CrossReference	Session
TimeUTC	Time-tag for scans.	Y	Scan	Scan
TimeUTC	Time-tag for stations	*Y*	*Station*	Station
TimeUTC	Time-tag for observations	Y	Observables	Obs
Tsys	System temperature		Observables	Obs
UVFperAsec	U V in FR per arcsec from CALC		Solve	Obs
WVR	WVR data		*Station*	Station

B.B Detailed Listing of vgosDB files.

Unless otherwise noted, all variables in a given file are required.

vgosDB File 1. AmbigSize_b?. Band dependent.

Variable Name	Description	Type	Dimension
AmbigSize	Size of ambiguities (seconds)	Real*8	(NumObs)

vgosDB File 2. AntennaApriori.nc

Variable Name	Description	Type	Dimension
AntennaName	Site names array.	Character*8	(NumStation)
AxisOffset	Axis offsets (m).	Real*8	(NumStation)
AxisTilt	Fixed axis tilt	Real*8	(2, NumStation)

AxisType	Axis type (1-eq 2-xy 3-azel 4 5)	Integer*4	(NumStation)

vgosDB File 3. AtmSetup.nc

Variable Name	Description	Type	Dimension
AtmRateSite	Site list for atm rate constraints	Character*8	(NumStation)
AtmRateConstraint	Atmosphere constraint. ps/hr	Real*8	(NumStation)
AtmInterval	Batchmode atmos interval - hours	Real*8	(NumStation)
GradOffsetConstraint	Gradient Offset Constraint	Real*8	(NumStation)
GradRateConstraint	Gradient Rate Constraint	Real*8	(NumStation)

vgosDB File 4. AzEl.nc

Variable Name	Description	Type	Dimension
AzTheo	Azimuth array definition	Real*8	(TimeDim2, NumStatScan)
ElTheo	Elevation array definition	Real*8	(TimeDim2, NumStatScan)

vgosDB File 5. Baseline.nc

Variable Name	Description	Type	Dimension
Baseline	Ref and rem site names.	Character*8	(2,NumObs)

vgosDB File 6. BaselineClockSetup.nc

Variable Name	Description	Type	Dimension
BaselineClockSite	List of baseline dependent clocks	Integer*4	See Note
BaselineClock	Bl-dependent clock list	Character*16	See Note

Used by solve. The dimenension is equal to the number of baseline dependent clocks.

vgosDB File 7. Cal-AxisOffset.nc

Variable Name	Description	Type	Dimension
AxisOffsetCal	New Axis Offset Contributions	Real*8	(TimeDim2, NumStatScan)

vgosDB File 8. Cal-Bend.nc

Variable Name	Description	Type	Dimension
BendCal	Consensus bending contrib. (sec)	Real*8	(TimeDim2,NumObs)

vgosDB File 9. Cal-BendSun.nc

Variable Name	Description	Type	Dimension
BendSunCal	Consensus bending contrib. (sec)	Real*8	(TimeDim2,NumObs)

vgosDB File 10. Cal-BendSunHigher.nc

Variable Name	Description	Type	Dimension
BendSunHigherCal	High order bending contrib.(sec)	Real*8	(TimeDim2,NumObs)

vgosDB File 11. Cal-Cable.nc

Variable Name	Description	Type	Dimension
CableCal	Cable calibration data	Real*8	(NumStatScan)
History	Description of cable-cal processing	Character*60	-
CableCalDataFlag	0=OK, -1=Missing, -2=bad, 1=Used Near pt	Integer*4	(NumStatScan)

vgosDB File 12. Cal-ClockApriori.nc

Variable Name	Description	Type	Dimension
ClockAprioriCal	Apriori clock contribution to delay, rate.	Real*8	(TimeDim2,NumObs)

vgosDB File 13. Cal-EarthTide.nc

Variable Name	Description	Type	Dimension
EarthTideCal	Earth tide contributions def.	Real*8	(TimeDim2,NumObs)

vgosDB File 14. Cal-EccentricityMap.nc

Variable Name	Description	Type	Dimension
EccentricityMapCal	Adding this maps the observables to the monment.	Real*8	(TimeDim2,NumObs)

vgosDB File 15. Cal-FeedCorrection.nc

Variable Name	Description	Type	Dimension
FeedCorrectionCal	Feedhorn corr. in CORFIL scheme	Real*8	(TimeDim2,NumObs)

vgosDB File 16. Cal-HFEOP-IERS2003.nc

Variable Name	Description	Type	Dimension
UT1OrthoCal	ORTHO_EOP Tidal UT1 contribution	Real*8	(TimeDim2,NumObs)
WobOrthoCal	ORTHO_EOP tidal wobble contribution	Real*8	(TimeDim2,NumObs)

vgosDB File 17. Cal-IonGroup_b?.nc Band dependent.

Variable Name	Description	Type	Dimension
IonGroupCal	Ion correction. Add to theo. Sec	Real*8	(TimeDim2,NumObs)
IonGroupCalDataFlag	0=OK, -1=Missing, -2=bad	Integer*4	(NumObs)
IonGroupCalSigma	Ion correction to sigma. Sec	Real*8	(TimeDim2,NumObs)

vgosDB File 18. Cal-IonPhase_b?.nc Band dependent.

Variable Name	Description	Type	Dimension
IonPhaseCal	Phase Corr Iono	Real*8	(NumObs)
IonPhaseCalSigma	Phase Corr RMS	Real*8	(NumObs)

vgosDB File 19. Cal-NMFDry.nc

Variable Name	Description	Type	Dimension
NMFDryCal	Nhmf (dry) atm. Contribution	Real*8	(TimeDim2, Num-

			StatScan)

vgosDB File 20. Cal-NMFWet.nc

Variable Name	Description	Type	Dimension
NMFWetCal	Whmf (wet) atm. Contribution	Real*8	(TimeDim2, NumStatScan)

vgosDB File 21. Cal-NutWahr.nc

Variable Name	Description	Type	Dimension
NutWahrCal	2000A Nut to Wahr Nut Contribution	Real*8	(TimeDim2,NumObs)

vgosDB File 22. Cal-OceanLoad.nc

Variable Name	Description	Type	Dimension
OceanLoadHorizCal	Site-dep ocean cont - horizontal	Real*8	(TimeDim2,NumStatScan)
OceanLoadVertCal	Site-dep ocean cont - vertical	Real*8	(TimeDim2,NumStatScan)
OceanLoadCal	Obs dependent ocean loading	Real*8	(TimeDim2,NumStatScan)

vgosDB File 23. Cal-Parallax.nc

Variable Name	Description	Type	Dimension
ParallaxCal	Parallax partial/contr 1 parsec	Real*8	(TimeDim2,NumObs)

vgosDB File 24. Cal-PoleTide.nc

Variable Name	Description	Type	Dimension
PoleTideCal	Pole tide contributions def.	Real*8	(TimeDim2,NumObs)

vgosDB File 25. Cal-PoleTideOldRestore.nc

Variable Name	Description	Type	Dimension
PoleTideOldRestoreCal	Old Pole Tide Restorer Contrib.	Real*8	(TimeDim2,NumObs)

vgosDB File 26. Cal-TiltRemover.nc

Variable Name	Description	Type	Dimension
TiltRemoverCal	Axis Tilt Contribution Remover	Real*8	(TimeDim2,NumObs)

vgosDB File 27. Cal-UnphaseCal.nc

Variable Name	Description	Type	Dimension
UnPhaseCal	UnPhaseCal effect - group&rate	Real*8	(2, TimeDim2,NumObs)

vgosDB File 28. Cal-Wobble.nc

Variable Name	Description	Type	Dimension
XwobbleCal	X Wobble contribution definition	Real*8	(TimeDim2,NumObs)
YwobbleCal	Y Wobble contribution definition	Real*8	(TimeDim2,NumObs)

vgosDB File 29. Cal-WobNut.nc

Variable Name	Description	Type	Dimension
WobNutCal	Short period nutation wobble con	Real*8	(TimeDim2,NumObs)

vgosDB File 30. CalcEop.nc

Variable Name	Description	Type	Dimension
CalcUt1Module	UT1 Module message definition	Character*80	-
CalcWobModule	Wobble message definition.	Character*80	-
UT1ArrayInfo	Array: (FJD of start, spacing in days, number points, Scaling (should be 1))	Real*8	(4)
UT1IntrpMode	Message for UT1 interp. scheme	Character*60	-
UT1Origin	Final Value TAI-UT1 origin text.	Character*80	-
UT1Values	Final Value TAI-UT1 data points.	Real*8	(15)
WobArrayInfo	Array: (FJD of start, spacing in days, number points)	Real*8	(3)
WobbleOrigin	Final Value wobble origin text.	Character*80	-
WobValues	Final wobble X Y component value	Real*8	(2,15)
WobIntrpMode	Interp. scheme for polar motion.	Character*60	-

This file contains information about the EOP values Calc used in its processing.

vgosDB File 31. CalcInfo.nc

Variable Name	Description	Type	Dimension
ATIFlag	ATIME Flow Control Message Def.	Character*80	-
ATIMessage	ATIME Message Definition	Character*80	-
ATMFlag	Atmosphere control flag mess def	Character*80	-
ATMMessage	Atmosphere message definition	Character*80	-
AxisOffsetFlag	Axis Offset Control flag mes def	Character*80	-
AxisOffsetMessage	Axis Offset Message Definition	Character*80	-
CalcFlagNames	CALC flow control flags name def	Character*4	(31)
CalcFlagValues	CALC flow control flags valu def	Integer*4	(31)
CalcVersion	CALC version number	Real*8	-
CoronaData	Corona model parameters.	Real*8	(2)
CTIFlag	CTIMG Flow Control Message Der	Character*80	-
CTIMessage	CTIMG Message Definition	Character*80	-
EarthTideData	Earth tide module data (la. h l)	Real*8	(3)
EarthTideFlag	Earth Tide flow control mess def	Character*80	-
EarthTideMessage	Earth Tide message definition	Character*80	-
FeedhornMessage	Feedhorn rot. angle mod. ident.	Character*80	-
NutationFlag	Nutation flow control mess def.	Character*80	-
NutationMessage	Nutation message definition	Character*80	-
OceanAmpEW	Ocean amp. E-W component(meters)	Real*8	(NumStation,2)

OceanAmpNS	Ocean amp. N-S component(meters)	Real*8	(NumStation,2)
OceanAmpUp	Ocean amp. radial comp.-meters.	Real*8	(NumStation,2)
OceanFlag	Ocean load flow control mess def	Character*80	-
OceanHorizontalAmp	Horz ocean loading ampltudes (m)	Real*8	(11,2,NumStation)
OceanHorizontalPhase	Horz ocean loading phases (rad).	Real*8	(11,2,NumStation)
OceanMessage	Ocean loading message definition	Character*80	-
OceanPhaseEW	Ocean phase E-W component (rad)	Real*8	(NumStation,2)
OceanPhaseNS	Ocean phase N-S component (rad)	Real*8	(NumStation,2)
OceanPhaseUp	Ocean phase radial comp.-radians	Real*8	(NumStation,2)
OceanStations	Ocean loading station status.	Character*4	(NumStation)
OceanUpAmp	Vert ocean loading ampltudes (m)	Real*8	(NumStation,11)
OceanUpPhase	Vert ocean loading phases (rad).	Real*8	(NumStation,11)
ParallaxFlag	Parallax flow control mess def	Character*80	-
ParallaxMessage	Parallax message definition	Character*80	-
PepMessage	PEP Utility Message Definition	Character*80	-
PoleTideFlag	Pole tide flow control mess def	Character*80	-
PoleTideMessage	Pole tide message definition	Character*80	-
PrecessionData	Precession constant (asec/cent).	Real*8	-
PrecessionFlag	Precession flow contril mess def	Character*80	-
PrecessionMessage	Precession message definition	Character*80	-
RelativityFlag	Relativisitc bending use status	Character*60	-
RelativityMessage	Relativity mod data (gamma).	Real*8	-
SiteMessage	Site Module Message Definition	Character*80	-
SiteZenithDelay	Site zenith path delays (nsec).	Real*8	(NumStation)
StarMessage	Star module message definition	Character*80	-
StarParallaxFlag	Parallax flow control mess def	Character*80	-
TheoryMessage	Theory module identification	Character*80	-
TidalUT1Flag	Flag for tidal terms in UT1 sers	Integer*4	-
UT1EPOCH	TAI - UT1 epoch value definition	Real*8	(2,2)
UT1Flag	UT1 control flag message def.	Character*80	-
WobbleFlag	Wobble flow control mess def.	Character*80	-
WOBEPOCH	Interpolated wobble array def	Real*8	(2,2)

vgosDB File 32. CalibrationSetup.nc

Variable Name	Description	Type	Dimension
CalStationName	List of sites for standard cal	Character*8	(NumStation)
FlybyFlag	Standard flcal configuration	Integer*4	(7,NumStation)
FlybyName	Key to the standard flcal config	Character*8	(8)
ObsCalFlag	Bit set indicate that calibration is recommended.	Integer*4	-
ObsCalName	Available obs dependent calibrations (poletide, earthdide, …)	Character*8	(14)

StatCalFlag	Bit set indicate that calibration is recommended.	Integer*4	(NumStation)
StatCalName	Station depedendent calibratipns (Cable, Phase, etc…)	Character*8	(6)

There are currently 14 observation dependent calibrations in solve. This number may change.

vgosDB File 33. ClockApriori.nc

Variable Name	Description	Type	Dimension
ClockAprioriRate	A priori clock drift (sec/sec)	Real*8	See note
ClockAprioriOffset	A priori clock offset (sec)	Real*8	See note
ClockAprioriSite	Stations with a priori clock mod	Character*8	See note

The ClockApriori.nc file is required only if there is a priori clock information. The dimension of the variables depends on the number of clocks with a priori information. Table 38 is a dump of the variable information for T2034 where there is information for two sites.

Table 38. Apriori clock information for T2034.

```
ClockAprioriSite =
  "GGAO7108",
  "MEDICINA" ;
 ClockAprioriOffset = 1, -0.00010329265 ;
 ClockAprioriRate = 0, 4.0358e-12 ;
```

vgosDB File 34. ClockBreak.nc

Variable Name	Description	Type	Dimension
BRK_NUMB	Number of batchmode clock breaks	Integer*4	-
CLKBREAK	Status of clock break existence (YE or NO)	Character*2	-
ClockBreakEpoch	Batchmode clock break epochs	Real*8	BRK_NUMB
ClockBreakFlag	Batchmode clock break flags	Integer*4	BRK_NUMB
ClockBreakSite	Sites with real clock breaks	Character*8	BRK_NUMB

vgosDB File 35. ClockSetup.nc

Variable Name	Description	Type	Dimension
CLK_CFLG	Clock constraint use flag. YE or NO	Character*2	-
ClockInterval	Batchmode clock interval – hours	Real*8	-
ClockRateConstraint	Clock constraint-Parts in 1.e14	Real*8	-
ClockRateName	Site list for clocks constraints	Character*8	(NumStation)
NumClockRefSite	# of clock reference stations	Integer*4	-
ReferenceClock	List of clock reference stations	Character*8	-

vgosDB File 36. Correlation_b?.nc Band dependent.

Variable Name	Description	Type	Dimension
Correlation	Corr coeff (0 --> 1).	Real*8	(NumObs)

vgosDB File 37. CorrInfo_bX.nc

Variable Name	Description	Type	Dimension
ABASACCE	Corel bas/apr accel (1/sec**2).	Real*8	(NumObs)
ABASDEL	Corel bas/apr delay (sec).	Real*8	(NumObs)
ABASRATE	Corel bas/apr delay rate (s/s).	Real*8	(NumObs)
APCLOFST	Apriori clock offset microsec.	Real*8	(NumObs)
AUTOEDIT	1=Run resulted from AUTOEDIT.	Integer*4	(NumObs)
BBCIndex	Physical BBC number by channel.	Integer*4	(2,NumChannels,NumObs)
BITSAMPL	Number of bits per sample.	Integer*4	(NumObs)
ChanAmpPhase	Amp(0-1) phs by chan(-180to180)	Integer*4	(2,NumChannels,NumObs)
ChannelFreq	RF freq by channel (MHz).	Real*8	(NumChannels, NumObs)
ChannelID	One-letter Fourfit channel ID.	Character*32	(NumObs)
CORBASCD	Correlator baseline code (2 ch).	Character*2	(NumObs)
CORBASNO	Correlator baseline number.	Integer*4	(NumObs)
CORCLOCK	Clock offset(sec)/rate(sec/sec).	Real*8	(2,2,NumObs)
CorCofErr	Corr. Coeff. formal error	Real*8	(NumObs)
CORELVER	Correlator software version numb	Integer*4	(NumObs)
CROOTFIL	Correlator root file name.	Character*16	(NumObs)
DBEDITVE	Dbedit revision date YYYY MM DD	Integer*4	(3,NumObs)
DELOBSVM	Obs delay at central epoch us.	Real*8	(2,NumObs)
DELRESID	Delay residual (sec).	Real*8	(NumObs)
DELTAEPO	Offset from center of scan (sec)	Real*8	(NumObs)
DISCARD	Percent data discarded by FRNGE.	Real*8	(NumObs)
DLYEPO+1	Phase delay at epoch+1 sec.	Real*8	(2,NumObs)
DLYEPO-1	Phase delay at epoch-1 sec.	Real*8	(2,NumObs)
DURATION	Scan duration (sec).	Real*8	(NumObs)
EffectiveDuration	Effective run duration sec.	Real*8	(NumObs)
ERRORATE	Log err rate by sta sb channel	Integer*4	(2,2,NumChannels,NumObs)
FOURFFIL	Fourfit output filename.	Character*16	(NumObs)
FOURFFXS	Fourfit output filename S-band	Character*16	(NumObs)
FOURFUTC	Fourfit processing time YMDHMS.	Integer*4	(6, NumObs)
FOURFVER	Fourfit version number.	Integer*4	(2, NumObs)
FRNGERR	Fourfit error flag blank=OK.	Character*2	(NumObs)
FRQGROUP	Frequency group code.	Character*2	(NumObs)
GeocMBD	Tot geocenter group delay (sec).	Real*8	(NumObs)
GeocPhase	Tot phase ref to cen of Earth.	Real*8	(NumObs)
GeocRate	Tot geocenter delay rate (s/s).	Real*8	(NumObs)
GeocResidPhase	Resid phs corrected to cen of E.	Real*8	(NumObs)
GeocSBD	Tot geocenter sbd delay (sec).	Real*8	(NumObs)
IDELAY	Corel instrumental delay (sec).	Real*8	(2, NumObs)

INCOH2	Incoh amp from FRNGE plot segs.	Real*8	(NumObs)
INCOHAMP	Fr. amp from incoh int of chan.	Real*8	(NumObs)
INDEXNUM	Corel index numbers by sb freq.	Integer*4	(2,16,NumObs)
LOFreq	LO frequencies per cha/sta MHz.	Real*8	(2,NumChannels, NumObs)
NumAccum	No. of accum. periods in Channel	Integer*4	(32, NumObs)
NumAp	# of AP by sideband and channel.	Integer*4	(2,NumChannels, NumObs)
NumBBCFreqs	No. of BBC frequencies in Band	Integer*4	(NumObs)
NumChannels	No. of U-L pairs in integration.	Integer*4	(NumObs)
NumSamples	# of samples by sideband and cha	Real*8	(2,NumChannels, NumObs)
OCCUPNUM	Site Occupation Number.	Character*8	(2,NumObs)
ORIGFILE	Original COREL file name.	Character*6	(NumObs)
PHASECAL	PC rate by sta (us per s).	Real*8	(2, NumObs)
PhaseCalAmpPhase	PC amp phs frq by sta channel.	Integer*2	(3,2,NumChannels,NumObs)
PhaseCalOffset	Phase cal offset (-18000/18000).	Integer*4	(2,NumChannels, NumObs)
Polarization	Polarization per sta/chan R/L.	Character*4	(NumChannels, NumObs)
ProbFalseDetection	Prob of false det from FRNGE.	Real*8	(NumObs)
QBFACTOR	Measure of uniformity of data.	Real*8	(NumObs)
RATOBSVM	Obs rate at central epoch .	Real*8	(NumObs)
RATRESID	Rate resid (sec per sec).	Real*8	(NumObs)
RECSETUP	Samp rate(kHz) Frames/PP PP/AP.	Integer*4	(3,NumObs)
RECTRACK	Trk table by sta sideb channel.	Integer*4	(2,2,14,NumObs)
REFCLKER	Ref sta clock epoch microsec.	Real*8	(NumObs)
RunCode	Run code e.g. "329-1300".	Character*8	(NumObs)
S2EFFREQ	Effective group freq for ion.	Real*8	(NumObs)
S2PHEFRQ	Effective phase frequency	Real*8	(NumObs)
S2REFREQ	Effective frequency for rate	Real*8	(NumObs)
SampleRate	Sample rate (Hz).	Real*8	(NumObs)
SBRESID	Single band delay residual.	Real*8	(NumObs)
SRCHPAR	FRNGE/Fourfit search parameters.	Real*8	(6,NumObs)
STARELEV	Elev angles calc by COREL.	Real*8	(2,NumObs)
StartOffset	Offset nominal start time (sec).	Integer*4	(NumObs)
STARTSEC	Start time in sec past hour.	Real*8	(NumObs)
StopOffset	Offset nominal stop time (sec).	Integer*4	(NumObs)
StopSec	Stop time in sec past hour.	Real*8	(NumObs)
TapeCode	Tape quality code.	Character*6	(NumObs)
TAPEID	Raw data tape ID for ref and rem	Character*8	(2,NumObs)
TimeSinceStart	Interval since start time (sec).	Real*8	(NumObs)
TotalFringeErr	Total fringe phase error (deg)	Real*8	(NumObs)
TOTPCENT	Tot phase at central epoch.	Real*8	(NumObs)
UTCCorr	UTC time tag of correlation.	Integer*4	(6,NumObs)

UTCErr	A priori UTC error site 1 (sec)	Real*8	(NumObs)
UTCMidObs	UTC at central epoch YMDHMS.	Integer*4	(6,NumObs)
UTCProc	YDDD of COREL by sta channel.	Integer*4	(2,14,NumObs)
UTCScan	Nominal scan time YMDHMS.	Integer*4	(6,NumObs)
UTCVLB2	UTC of FRNGE proc YMDHMS.	Integer*4	(6,NumObs)
VFDWELL	Dwell time in each channel (sec)	Real*8	(NumObs)
VFRQAM	Normalized channel amplitude	Real*8	(32,NumObs)
VFRQPCAM	Phase cal tone Amplitudes	Real*8	(2,32,NumObs)
VFRQPCFR	Phase cal tone Frequencies	Real*8	(2,32,NumObs)
VFRQPCPH	PHASE cal tone Phases	Real*8	(2,32,NumObs)
VFRQPH	Channel Phase (degrees)	Real*8	(32,NumObs)
VIRTFREQ	Sky Frequencies	Real*8	(32,NumObs)
VLB1FILE	Correlator file name.	Character*6	(NumObs)
VLB1XTNT	corr. ext by sideb channel.	Integer*4	(NumObs)
VLB2PRG	FRNGE(YYMMDD) Fourfit(x.x) ver.	Character*6	(NumObs)
VLB2XTNT	FRNGE extent number.	Integer*4	(NumObs)
URVR	Rate derivatives mHz per asec.	Real*8	(2,NumObs)
ZDELAY	Corel zenith atmos. delay (sec).	Real*8	(2,NumObs)

The vgosDB file CorrInfo.nc contains information about the correlation and fringing process and other information about the session. This information is seldom used but is kept for archival purposes. The exact data in this file depends on the particular correlator used and the version of the correlator software. Software developers are encouraged to dump as much information as they think may someday be used. They are also encouraged to use as many of the above vgosDB variables which make sense, adding new ones as appropriate.

vgosDB File 38. DelayDataFlag_b?.nc Band dependent.

Variable Name	Description	Type	Dimension
DelayDataFlag	0=OK, -1=Missing, -2=bad,-3=sigma small, -4=sigma big	Integer*4	(NumObs)

vgosDB File 39. DelayTheoretical.nc

Variable Name	Description	Type	Dimension
DelayTheoretical	Consensus theoretical delay	Real*8	(NumObs)

vgosDB File 40. Dis_kOceanLoad.nc

Variable Name	Description	Type	Dimension
OceanLoadDis	Ocean load site dependent displace	Real*8	(3,TimeDim2,NumStatScan)

vgosDB File 41. EccentricityApriori.nc

Variable Name	Description	Type	Dimension
EccentricityName	Site names array.	Character*8	(NumStation)

EccentricityMonument	Eccentricity monument name	Character*18	(Numstation)
EccentricityType	Eccentricity type: XY or NE	Character*2	(NumStation)
EccentricityVector	Eccentricity taken from eccentricity file.	Real*8	(3,NumStation)

vgosDB File 42. Edit.nc

Variable Name	Description	Type	Dimension
DelayFlag	Delay unweight flag	Integer*4	(NumObs)
PhaseFlag	Phase unweight flag	Integer*4	(NumObs)
RateFlag	Delay rate unweight flag.	Integer*4	(NumObs)
UserSup	User action for suppression	Integer*4	(NumObs)

vgosDB File 43. EffFreq_b?.nc Band dependent

Variable Name	Description	Type	Dimension
FreqGroupIon	Effective Group Delay Ionospheric Frequency	Real*8	(NumObs)
FreqPhaseIon	Effective Phase Delay Ionospheric Frequency	Real*8	(NumObs)
FreqRateIon	Effective Group Rate Ionospheric Frequency	Real*8	(NumObs)

vgosDB File 44. EffFreq_kNoAmpWt_b?.nc Band dependent

Variable Name	Description	Type	Dimension
FreqGroupIon	Effective Group Delay Ionospheric Frequency. All channels equal wt.	Real*8	(NumObs)
FreqPhaseIon	Effective Phase Delay Ionospheric Frequency. All channels equal wt.	Real*8	(NumObs)
FreqRateIon	Effective Group Rate Ionospheric Frequency. All channels equal wt.	Real*8	(NumObs)

vgosDB File 45. EopApriori.nc

Variable Name	Description	Type	Dimension
PolarMotion	Polar motion X & Y for obs (rad)	Real*8	(2, NumScan)
UT1	UT1 time of day for this obsvr.	Real*8	(NumScan)

vgosDB File 46. EopSetup.nc

Variable Name	Description	Type	Dimension
UT1OffsetConstraint	UT1 Offset Constraint	Real*8	-
WobOffsetConstraint	Polar Motion Offset Constraint	Real*8	-

vgosDB File 47. Ephemeris_kDE405JPL.nc

Variable Name	Description	Type	Dimension
EarthXYZ	Earth barycentric coordinates.	Real*8	(3, TimeDim3,NumScan)
MoonXYZ	Lunar geocentric coordinates.	Real*8	(3, TimeDim2, NumScan)
SunXYZ	Solar geocentric coordinates.	Real*8	(3, TimeDim2, NumScan)

vgosDB File 48. Head.nc

Variable Name	Description	Type	Dimension	Required?
Correlator	Correlator name.	Character*32	-	N
CorrelatorType	Correlator type: MK3/MK4/K4 etc.	Character* 8	-	N
ExpDescription	Experiment description.	Character*80	-	N
ExpName	Experiment name.	Character*16	-	Y
ExpSerialNumber	Experiment Serial Number.	Integer*4	-	N
iUTCInterval	First and last UTC time tag in input file.	Integer*2	(5,2)	Y
NumObs	Number of observations (I*4)	Integer*4	-	Y
NumScan	Number of Scans (Integer*4)	Integer*4	-	Y
NumSource	Number of radio sources.	Integer*2	-	Y
NumStation	Number of sites.	Integer*2	-	Y
PrincipalInvestigator	Agency/contact_person/PI name.	Character*80	-	N
RecordingMode	Recoding mode.	Character*80	-	N
SourceList	Source names array.	Character*8	(NumSource)	Y
StationList	Site names array.	Character*8	(NumStation)	Y

vgosDB File 49. FeedRotation.nc

Variable Name	Description	Type	Dimension
FeedRotation	Feedhorn rotation angle.	Real*8	(NumStatScan)

vgosDB File 50. FeedRotNet.nc

Variable Name	Description	Type	Dimension
FeedRotNet	Feedhorn correction for phase delay= FeedRotSite1-FeedRotSite2	Real*8	(NumObs)

vgosDB File 51. FractC.nc

Variable Name	Description	Type	Dimension
FractC	Coordinate time at site 1	Real*8	(NumObs)

vgosDB File 52. GroupBLWeights.nc

Variable Name	Description	Type	Dimension
GroupBLWeightName	B.L.names for formal errors	Character*8	(2, XXX)
GroupBLWeights	Group delay and rate re-weighting constants.	Real*8	(XXX, 2)

The XXX dimension depends on the number of baseline weights. Usually this is the number of baselines.

vgosDB File 53. GroupDelay_b?.nc Band dependent

Variable Name	Description	Type	Dimension
GroupDelay	Delay observable produced by fringing.	Real*8	(NumObs)
GroupDelaySig	Delay Measurement Sigma	Real*8	(NumObs)

vgosDB File 54. GroupDelayFull_b?.nc Band dependent

Variable Name	Description	Type	Dimension
GroupDelayFull	Delay Observable with ambiguities resolved and added.	Real*8	(NumObs)

vgosDB File 55. GroupRate_b?.nc Band dependent

Variable Name	Description	Type	Dimension
GroupRate	Rate Observable	Real*8	(NumObs)
DataFlag	0=OK, -1=Missing, -2=bad,-3=sigma small, -4=sigma big	Integer*4	(NumObs)
GroupRateSig	Rate Measurement Sigma	Real*8	(NumObs)

vgosDB File 56. IonBits.nc

Variable Name	Description	Type	Dimension
IonBits	ICORR for full ion tracking.	Integer*4	(NumObs)

vgosDB File 57. IonSetup.nc

Variable Name	Description	Type	Dimension
IonSolveFlag	Bit flag indicating station has iono correction	Integer*4	(NumStation)
IonStations	Stations with ionocorrection	Character*8	(NumStation)

vgosDB File 58. Met.nc

Variable Name	Description	Type	Dimension
AtmPres	Pressure in hPa at site	Real*8	(NumStatScan)
RelHum	Rel.Hum. at local WX st (50%=.5)	Real*8	(NumStatScan)
TempC	Temp in C at local WX station	Real*8	(NumStatScan)

vgosDB File 59. Misc.nc

Variable Name	Description	Type	Dimension
Aplength	Length of accumul. period in sec	Real*8	
CableSign	Signs of cable cal application	Character*9	-
StationNameCable	Stations for cable sign	Character*8	(NumStation)
FourfitControlFile	Control file name for fourfit.	Character*	
FourFitCmdCString	Command string used for fourfit.	Character*	
NumLagsUsed	Num of lags used for correlation	Integer*4	
LeapSecond	Leap second	Real*8	-

vgosDB File 60. NGRADPAR.nc

Variable Name	Description	Type	Dimension
NMFGradPart	Niell dry atm. gradient partials	Real*8	(2,2,NumObs)

vgosDB File 61. NGSQualityFlag.nc

Variable Name	Description	Type	Dimension
NGSQualityFlag	Zero means good. A Combination of Delay-Flag and IonCode	Integer*4	(NumObs)

vgosDB File 62. NumGroupAmbig_b?.nc Band dependent

Variable Name	Description	Type	Dimension
NumGroupAmbig	Number of group delay ambiguities.	Integer*4	(NumObs)

vgosDB File 63. NumPhaseAmbig_b?.nc Band dependent

Variable Name	Description	Type	Dimension
NumPhaseAmbig	Number of phase ambiguities.	Integer*4	(NumObs)

vgosDB File 64. Nut_k2000PsiEps.nc

Variable Name	Description	Type	Dimension
Nut2000PsiEps	IAU200A Nut. - Dpsi Deps Rates	Real*8	(2,TimeDim2, NumScans)

vgosDB File 65. Nut_k2000XYS.nc

Variable Name	Description	Type	Dimension
Nut2000XYS	CIP Coordinates X Y S and Rates	Real*8	(3,TimeDim2, NumScans)

vgosDB File 66. Nut_kWahr.nc

Variable Name	Description	Type	Dimension
NutWahrPsiEps	Wahr nut vals - Dpsi Deps&rates	Real*8	(2, TimeDim2,NumScans)

vgosDB File 67. ObsCrossRef.nc

Variable Name	Description	Type	Dimension	Required?
Obs2Baseline	Cross reference from observation to baseline. Stations assumed alphabetical.	Integer*4	(2,NumObs)	Y
Obs2Scan	Cross reference from observation to scan	Integer*4	(NumObs)	Y

vgosDB File 68. Part_ kAxisOffset.nc

Variable Name	Description	Type	Dimension
AxisOffsetPart	Axis Offset partial deriv. def.	Real*8	(TimeDim2, NumObs)

vgosDB File 69. Part-Bend.nc

Variable Name	Description	Type	Dimension
BendPart	Grav. bend. partial w.r.t. Gamma	Real*8	(TimeDim2, NumObs)

vgosDB File 70. Part-EOP.nc

Variable Name	Description	Type	Dimension
UT1Part	UT1 partial derivatives def.	Real*8	(2,TimeDim2, NumObs)
WobblePart	Wobble partial derivatives def.	Real*8	(2,TimeDim2, NumObs)

vgosDB File 71. Part-Gamma.nc

Variable Name	Description	Type	Dimension
GammaPart	Consensus partial w.r.t. Gamma	Real*8	(TimeDim2, NumObs)

vgosDB File 72. Part-NMFDry.nc

Variable Name	Description	Type	Dimension
NMFDryPart	Nhmf2 dry partial deriv. def.	Real*8	(TimeDim2, NumObs)

vgosDB File 73. Part-NMFGrad.nc

Variable Name	Description	Type	Dimension
NMFGradEWPart	NMF dry atm. EW Gradient Partials	Real*8	(TimeDim2, NumObs)
NMFGradNSPart	NMF dry atm. NS Gradient Partials	Real*8	(TimeDim2, NumObs)

vgosDB File 74. Part-NMFWet.nc

Variable Name	Description	Type	Dimension
NMFWetPart	NMF Wet Partial	Real*8	(TimeDim2, NumObs)

vgosDB File 75. Part-Nut2KPsiEps.nc

Variable Name	Description	Type	Dimension
Nut2000PsiEpsPart	IAU2000A Nutation Psi Eps Partls	Real*8	(2,TimeDim2, NumObs)

vgosDB File 76. Part-Nut2KXY.nc

Variable Name	Description	Type	Dimension
Nut2KXYPart	IAU2000A Nutation X Y Partials	Real*8	(2,TimeDim2, NumObs)

vgosDB File 77. Part-NutPsiEps.nc

Variable Name	Description	Type	Dimension
NutPsiEpsPart	Pre Calc10 Nuation Partial	Real*8	(2,TimeDim2, NumObs)

vgosDB File 78. Part-Parallax.nc

Variable Name	Description	Type	Dimension
ParallaxPart	Parallax partial deriv. def.	Real*8	(TimeDim2, NumObs)

vgosDB File 79. Part-Poletide.nc

Variable Name	Description	Type	Dimension
WobPart	Pole Tide Partials w.r.t. X & Y	Real*8	(2,TimeDim2, NumObs)

vgosDB File 80. Part-Precession.nc

Variable Name	Description	Type	Dimension
PrecessionPart	Precession partial deriv. def.	Real*8	(TimeDim2, NumObs)

vgosDB File 81. Part-RaDec.nc

Variable Name	Description	Type	Dimension

RaDecPart	Star partial derivatives def.	Real*8	(2,TimeDim2, NumObs)

vgosDB File 82. Part-XYZ.nc

Variable Name	Description	Type	Dimension
XYZPart	Site partials: dtau/dr_1=-dtau/dr_2	Real*8	(3,TimeDim2, NumObs)

vgosDB File 83. Phase_b?.nc Band dependent

Variable Name	Description	Type	Dimension
Phase	Total phase.	Real*8	(NumObs)
PhaseDataFlag	0=OK, -1=Missing, -2=bad,-3=sigma small, -4=sigma big	Integer*4	(NumObs)
PhaseSig	Phase delay sigma.	Real*8	(NumObs)

vgosDB File 84. PhaseBLWeights.nc

Variable Name	Description	Type	Dimension
PhaseBLWeights	Phase formal error constants.	Real*8	(2, NumBaselines)

vgosDB File 85. PhaseDelayFull_b?.nc Band dependent

Variable Name	Description	Type	Dimension
PhaseDelayFull	Phase delay resolved with ambiguities added.	Real*8	(NumObs)

vgosDB File 86. QualityCode_b?.nc Band dependent

Variable Name	Description	Type	Dimension
QualityCode	FRNGE quality index 0 --> 9	Character*1	(NumObs)

vgosDB File 87. RateTheoretical.nc

Variable Name	Description	Type	Dimension
RateTheoretical	Consensus theoretical rate.	Real*8	(NumObs)

vgosDB File 88. RefFreq_b?.nc Band dependent

Variable Name	Description	Type	Dimension
RefFreq	Frequency to which phase is referenced.	Real*8	-

vgosDB File 89. Rot_kCF2J2K.nc

Variable Name	Description	Type	Dimension
CF2J2K	Crust-fixed to J2000 Rot. Matrix and deriviatives.	Real*8	(3,3,TimeDim3, NumScans)

vgosDB File 90. SBDelay_b?.nc Band dependent

Variable Name	Description	Type	Dimension
SBDelay	Single band delay	Real*8	(NumObs)
SBDelaySig	Single band delay error	Real*8	(NumObs)

vgosDB File 91. ScanCrossRef.nc

Variable Name	Description	Type	Dimension	Required?
Scan2Source	Cross reference scan to source	Integer*4	(NumScans)	Y
SourceNameCrossRef	Source names. (Character*8)	Character*8	(NumSource)	Y

vgosDB File 92. ScanName.nc

Variable Name	Description	Type	Dimension
ScanName	Scanname in database	Character*10	(NumScans)
ScanNameFull	ScanName=UTC TimeTag + SourceName	Character*30	(NumScans)

Below is a partial dump of this file to illustrate the format.

Table 39. Partial Dump of ScanName.nc

```
ScanName =
  "004-1700  ",
  "004-1701  ",
  "004-1702  ",
  "004-1705  ",
  "004-1708a ",
 ….
ScanNameFull =
  "2010/01/04-17:00:20.0 0014+813",
  "2010/01/04-17:02:18.0 1636+473",
  "2010/01/04-17:03:03.0 1334-127",
  "2010/01/04-17:06:25.0 3C418   ",
```

vgosDB File 93. ScanTimeMJD.nc

Variable Name	Description	Type	Dimension
DayFrac	Fractional day time tag of scan	Real*8	(NumScans)
MJD	MJD time tag of scan	Integer*4	(NumScans)

vgosDB File 94. SelectionStatus.nc

Variable Name	Description	Type	Dimension
BaselineSelectionFlag	Baseline selection bit maped array. 1=some obs, etc.	Integer*4	(NumStation, NumStation)
SourceSelectionFlag	Source selection status bit-mapped array.	Integer*4	(NumSource)

vgosDB File 95. SNR_b?.nc Band dependent

Variable Name	Description	Type	Dimension
SNR	Signal to noise ratio.	Real*8	(NumObs)

vgosDB File 96. Source.nc

Variable Name	Description	Type	Dimension
Source	Radio source name.	Character*8	(NumObs)

vgosDB File 97. SourceApriori.nc

Variable Name	Description	Type	Dimension	Required?
Source2000RaDec	J2000 Source RA and Dec (Radians)	Real*8	(2, NumSource)	Y
SourceNameApriori	Source names in RA order.	Character*8	(NumSource)	Y
SourceReference	Source of coordinate values.	Character*20	(NumSource)	N But recommneded

vgosDB File 98. SourceCrossRef.nc

Variable Name	Description	Type	Dimension
Scan2Source	Cross reference scan to source	Integer*4	(NumScans)
SourceNameCrossRef	Source names.	Character*8	(NumSource)

vgosDB File 99. StationApriori.nc

Variable Name	Description	Type	Dimension	Required?
StationNameApriori	Site names in alphabetical order	Character*8	(NumStation)	Y
StationXYZ	Site cartesian coords (m).	Real*8	(3, NumStation)	Y

vgosDB File 100. Station CrossRef.nc

Variable Name	Description	Type	Dimension
NumScansPerStation	Number of scans per station.	Integer*4	(NumStation)
Scan2Station	Cross reference scans to station	Integer*4	(NumStation, NumScans)
Station2Scan	Cross reference station-scan to schedule-scan	Integer*4	(NumStation, NumScans)
StationNameCrossRef	Site names in alphabetical order. Char*8	Character*8	(NumStation)

vgosDB File 101. TimeUTC.nc (for scans)

Variable Name	Description	Type	Dimension
Second	Seconds part of time tag for scan	Real*8	(NumScans)
YMDHM	YMDHM time tag of scan	Integer*4	(5, NumScans)

vgosDB File 102. TimeUTC.nc (for stations)

Variable Name	Description	Type	Dimension
Second	Seconds part of time-tag	Real*8	(NumStatScan)
YMDHM	YMDHM time tag of station-scan	Integer*4	(5, NumStatScan)

vgosDB File 103. TimeUTC.nc (for observations)

Variable Name	Description	Type	Dimension
Second	Seconds part of UTC TAG.	Real*8	(NumObs)
YMDHM	YMDHM timetag of observation.	Integer*4	(5, NumObs)

vgosDB File 104. Tsys.nc

Variable Name	Description	Type	Dimension
Tsys	TSYS TMP (K) 14CH AND IF1 IF2	Real*8	(NumChannels,NumScans

Only a few files have Tsys.nc.

vgosDB File 105. UnPhaseCalFlag.nc

Variable Name	Description	Type	Dimension
UNPHAFLG	UnPhaseCal flag	Integer*4	(2,NumObs)
			(2,NumObs)

vgosDB File 106. UVFperAsec.nc

Variable Name	Description	Type	Dimension
UVFperAsec	UV in FR per arcsec from CALC	Real*8	(2,NumObs)

vgosDB File 107. WVR.nc

Variable Name	Description	Type	Dimension
F12VOLT	F1, F2 VOLT .	Real*8	(2,NumStatScan
WVRDelay	WVR LINE OF SIGHT & ZENITH(NSEC)	Real*8	(2,NumStatScan)
WVRQcode	WVR DELAY DATA QUALITY CODE	Integer*4	(2,NumStatScan)
WVRTemp	BRT TEMP (K) OF WVR ,F1,F2.	Real*8	(2,NumStatScan)
WVRTempRef	REFERENCE1,2 WVR TEMP (K).	Real*8	(3,NumStatScan)

The variables included in WVR.nc depend on the the kind of WVR instrument. The only required variable is WVRDelay.

C Complete List of vgosDB Variables

This section gives a complete list of all of the vgosDB variables and their scope. vgosDB variable names are unique once you specify the station or band.

The table below gives the dimensions when they are fixed or can be determined in advance. A few dimensions can vary from database to database. For example, the dimesions of variables associated with the a priori clock depend on the number of stations we have a priori clock information for. In these cases you should should refer to the documentation for the particular file.

Table 40. Complete list of vgosDB Variables

Name	Description	Scope	Directory	File	Type	Dim
ABASACCE	Corel bas/apr accel (1/sec**2).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
ABASDEL	Corel bas/apr delay (sec).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
ABASRATE	Corel bas/apr delay rate (s/s).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
AmbigSize	Group delay ambiguity spacing	Obs	Observables	AmbigSize_b?	Real*8	(NumObs)
AntennaName	Site names array.	Session	Apriori	AntennaApriori	Char*8	(NumStation)
APCLOFST	Apriori clock offset microsec.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
Aplength	Length of accumul. period in sec	Session	Solve	Misc	Real*8	-
ATIFlag	ATIME Flow Control Message Def.	Session	ObsTheoretical	CalcInfo	Char*80	-
ATIMessage	ATIME Message Definition	Session	ObsTheoretical	CalcInfo	Char*80	-
ATMFlag	Atmosphere control flag mess def	Session	ObsTheoretical	CalcInfo	Char*80	-
AtmInterval	Batchmode atmos interval - hours	Session	Solve	AtmSetup	Real*8	(NumStat)
ATMMessage	Atmosphere message definition	Session	ObsTheoretical	CalcInfo	Char*80	-
AtmPres	Pressure in hPa at site	Station	-	Met	Real*8	(NumStatScan)
AtmRateConstraint	Atmosphere constraint. ps/hr	Session	Solve	AtmSetup	Real*8	(NumStat)
AtmRateSite	Site list for clocks constraints	Session	Solve	AtmSetup	Char*8	(NumStat)
AUTOEDIT	1=Run resulted from AUTOEDIT.	Obs	Observables	CorrInfo_b?	Integer*4	(NumObs)

AxisOffset	Axis offsets (m).	Session	Apriori	AntennaApriori	Real*8	(NumStation)
AxisOffsetCal	New Axis Offset Contributions	Station	-	Cal-AxisOffset	Real*8	(TimeDim2, Num-StatScan)
AxisOffsetFlag	Axis Offset Control flag mes def	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
AxisOffsetMessage	Axis Offset Message Definition	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
AxisOffsetPart	Axis Offset partial deriv. def.	Station	-	Part-AxisOffset	Real*8	(TimeDim2,NumObs)
AxisTilt	Fixed axis tilt	Session	Apriori	AntennaApriori	Real*8	(2, NumStation)
AxisType	Axis type (1-eq 2-xy 3-azel 4 5)	Session	Apriori	AntennaApriori	Inte-ger*4	(NumStation)
AzTheo	Azimuth array definition	Station	-	AzEl	Real*8	(TimeDim2, Num-StatScan)
Baseline	Ref and rem site names.	OBs	Observables	Baseline	Char*8	(2,NumObs)
BaselineClock	List of baseline dependent clocks	Session	Solve	BaselineClockSetup	Inte-ger*4	(NumBslnClock)
BaselineSelectionFlag	Baseline selection bit maped array. 1=some obs, etc.	Session	Solve	SelectionStatus	Inte-ger*4	(NumStation ,NumStation)
BBCIndex	Physical BBC number by channel.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(2,NumChannels,NumObs)
BendCal	Consensus bending contrib. (sec)	Obs	ObsCalTheo	Cal-Bend	Real*8	(TimeDim2,NumObs)
BendPart	Grav. bend. partial w.r.t. Gamma	Obs	ObsPart	Part-Bend	Real*8	(TimeDim2,NumObs)
BendSunCal	Consensus bending contrib. (sec)	Obs	ObsCalTheo	Cal-BendSun	Real*8	(TimeDim2,NumObs)
BendSunHigherCal	High order bending contrib.(sec)	Obs	ObsCalTheo	Cal-BendSunHigher	Real*8	(TimeDim2,NumObs)
BITSAMPL	Number of bits per sample.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(NumObs)
BRK_NUMB	Number of batchmode clock breaks	Session	Session	ClockBreak	Inte-ger*4	-
CableCal	Cable calibration data	Station	-	Cal-Cable	Real*8	(NumStatScan)
CableCalDataFlag	0=OK, -1=Missing, -2=bad, 1=Used Near pt	Station	-	Cal-Cable	Inte-ger*4	(NumStatScan)
CableSign	Signs of cable cal application	Session	Solve	Misc	Char*9	-
CalcFlagNames	CALC flow control flags name def	Session	ObsTheoreti-cal	CalcInfo	Char*4	-31

CalcFlagValues	CALC flow control flags valu def	Session	ObsTheoretical	CalcInfo	Integer*4	-31
CalcUt1Module	UT1 Module message definition	Session	Solve	CalcEop	Char*80	-
CalcVersion	CALC version number	Session	ObsTheoretical	CalcInfo	Real*8	-
CalcWobModule	Wobble message definition.	Session	Solve	CalcEop	Char*80	-
CalStationName	List of sites for standard cal	Session	Solve	CalibrationSetup	Char*8	(NumStation)
CF2J2K	Crust-fixed to J2000 Rot. Matrix and deriviatives	Scan	Scan	Rot_kCF2J2K	Real*8	(3,3,TimeDim3, NumScan)
ChanAmpPhase	Amp(0-1) phs by chan(-180to180)	Obs	Observables	CorrInfo_b?	Real*8	(2,NumChannels,NumObs)
ChannelFreq	RF freq by channel (MHz).	Obs	Observables	CorrInfo_b?	Real*8	(NumChannels,NumObs)
ChannelID	One-letter Fourfit channel ID.	Obs	Observables	CorrInfo_b?	Char*32	(NumObs)
CLK_CFLG	Clock constraint use flag.	Session	Solve	ClockSetup	Char*2	-
CLKBREAK	Status of clock break existence	Session	Session	ClockBreak	Char*2	-
ClockAprioriCal	Apriori clock contribution to delay, rate.	Obs	ObsCalTheo	Cal-ClockApriori	Real*8	(TimeDim2,NumObs)
ClockAprioriOffset	A priori clock offset (sec)	Session	Apriori	ClockApriori	Real*8	See file
ClockAprioriRate	A priori clock drift (sec/sec)	Session	Apriori	ClockApriori	Real*8	See file
ClockAprioriSite	Stations with a priori clock mod	Session	Apriori	ClockApriori	Char*8	See file
ClockBreakEpoch	Batchmode clock break epochs	Session	Session	ClockBreak	Real*8	BRK_NUMB
ClockBreakFlag	Batchmode clock break flags	Session	Session	ClockBreak	Integer*4	BRK_NUMB
ClockBreakSite	Sites with real clock breaks	Session	Session	ClockBreak	Char*8	BRK_NUMB
ClockInterval	Batchmode clock interval - hours	Session	Solve	ClockSetup	Real*8	-
ClockRateConstraint	Clock constraint-Parts in 1.e14	Session	Solve	ClockSetup	Real*8	-
ClockRateName	Site list for clocks constraints	Session	Solve	ClockSetup	Char*8	(NumStation)
CORBASCD	Correlator baseline code (2 ch).	Obs	Observables	CorrInfo_b?	Char*2	(NumObs)
CORBASNO	Correlator baseline number.	Obs	Observables	CorrInfo_b?	Integer*4	(NumObs)
CORCLOCK	Clock offset(sec)/rate(sec/sec).	Obs	Observables	CorrInfo_b?	Real*8	(2,2,NumObs)
CorCofErr	Corr. Coeff. formal error	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)

CORELVER	Correlator software version numb	Obs	Observables	CorrInfo_b?	Integer*4	(NumObs)
CoronaData	Corona model parameters.	Session	ObsTheoretical	CalcInfo	Real*8	(2)
Correlation	Corr coeff (0 --> 1) for S-band	Obs	Observables	Correlation_b?	Real*8	(NumObs)
Correlator	Correlator name.	Session	-	Head	Char*32	-
CorrelatorType	Correlator type: MK3/MK4/K4 etc.	Session	-	Head	Char*8	-
CROOTFIL	Correlator root file name.	Obs	Observables	CorrInfo_b?	Char*60	(NumObs)
CTIFlag	CTIMG Flow Control Message Der	Session	ObsTheoretical	CalcInfo	Char*80	-
CTIMessage	CTIMG Message Definition	Session	ObsTheoretical	CalcInfo	Char*80	-
DataFlag	0=OK, -1=Missing, -2=bad,-3=sigma small, -4=sigma big	Obs	Observables	GroupRate_b?	Integer*4	(NumObs)
DayFrac	Fractional day time tag of scan	Scan	Solve	ScanTimeMJD	Real*8	(NumScan)
DBEDITVE	Dbedit revision date YYYY MM DD	Obs	Observables	CorrInfo_b?	Integer*4	(3,NumObs)
DelayDataFlag	0=OK, -1=Missing, -2=bad,-3=sigma small, -4=sigma big	Obs	Observables	DelayDataFlag_b?	Integer*4	(NumObs)
DelayFlag	Delay unweight flag	Obs	ObsEdit	Edit	Integer*4	(NumObs)
DelayTheoretical	Consensus theoretical delay	Obs	ObsTheoretical	DelayTheoretical	Real*8	(NumObs)
DELOBSVM	Obs delay at central epoch us.	Obs	Observables	CorrInfo_b?	Real*8	(2,NumObs)
DELRESID	Delay residual (sec).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
DELTAEPO	Offset from center of scan (sec)	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
DISCARD	Percent data discarded by FRNGE.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
DLYEPO+1	Phase delay at epoch+1 sec.	Obs	Observables	CorrInfo_b?	Real*8	(2,NumObs)
DLYEPO-1	Phase delay at epoch-1 sec.	Obs	Observables	CorrInfo_b?	Real*8	(2, NumObs)
DomesNumber	Domes number array	Session	Apriori	StationApriori	Char*8	(NumStation)
DURATION	Scan duration (sec).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
EarthTideCal	Earth tide contributions def.	Obs	ObsCalTheo	Cal-EarthTide	Real*8	(2,NumObs)
EarthTideData	Earth tide module data (la. h l)	Session	ObsTheoreti-	CalcInfo	Real*8	(3)

			cal			
EarthTideFlag	Earth Tide flow control mess def	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
EarthTideMessage	Earth Tide message definition	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
EarthXYZ	Earth barycentric coordinates.	Scan	Scan	Ephemer-is_kDE405JPL	Real*8	(3, TimeDim3, Num-Scan)
EccentricityMapCal	Adding this maps the observables to the monument.	Obs	ObsCalTheo	Cal-EccentricityMap	Real*8	(TimeDim2,NumObs)
EccentricityMonu-ment	Eccentricity monument name	Session	Apriori	EccentricityApriori	Char*18	(NumStation)
EccentricityName	Site names array.	Session	Apriori	EccentricityApriori	Char*8	(NumStation)
EccentricityType	Eccentricity type: XY or NE	Session	Apriori	EccentricityApriori	Char*2	(NumStation)
EccentricityVector	Eccentricity taken from eccentricity file.	Session	Apriori	EccentricityApriori	Real*8	(3,NumStation)
EffectiveDuration	Effective run duration sec.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
ElTheo	Elevation array definition	Station	-	AzEl	Real*8	(TimeDim2,NumStatSca n)
ERRORATE	Log err rate by sta sb channel	Obs	Observables	CorrInfo_b?	Inte-ger*4	(2,2,NumChannels,Num Obs)
ExpDescription	Experiment description.	Session	-	Head	Char*82	-
ExpName	Observing program (exp.) name	Session	-	Head	Char*32	-
ExpSerialNumber	Experiment Serial Number.	Session	-	Head	Inte-ger*4	-
F12VOLT	F1, F2 VOLT .	Station	-	WVR	Real*8	(2,2)
FeedCorrectionCal	Feedhorn corr. in CORFIL scheme	Obs	ObsCalTheo	Cal-FeedCorrection	Real*8	(TimeDim2,NumObs)
FeedhornMessage	Feedhorn rot. angle mod. ident.	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
FeedRotation	Feedhorn rotation angle.	Station	-	FeedRotation	Real*8	(NumStatScan)
FeedRotNet	Feedhorn correction for phase delay= FeedRotSite1-FeedRotSite2	Obs	ObsDerived	FeedRotNet	Real*8	(NumObs)
FlybyFlag	Standard flcal configuration	Session	Solve	CalibrationSetup	Inte-ger*4	(7,NumStation)
FlybyName	Key to the standard flcal config	Session	Solve	CalibrationSetup	Char*8	(8)
FOURFFIL	Fourfit output filename.	Obs	Observables	CorrInfo_b?	Char*16	(NumObs)

FOURFFXS	Fourfit output filename S-band	Obs	Observables	CorrInfo_b?	Char*16	(6, NumObs)
FourFitCmdCString	Command string used for fourfit.	Session	Solve	Misc	Char*12 8	-
FourfitControlFile	Control file name for fourfit.	Session	Solve	Misc	Char*96	-
FOURFUTC	Fourfit processing time YMDHMS.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(2, NumObs)
FOURFVER	Fourfit version number.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(NumObs)
FractC	Coordinate time at site 1	Obs	Solve	FractC	Real*8	(NumObs)
FreqGroupIon	Effective Group Delay Ionospheric Fre-quency	Obs	ObsDerived	EffFreq_b?	Real*8	(NumObs)
FreqPhaseIon	Effective Phase Delay Ionospheric Fre-quency	Obs	ObsDerived	EffFreq_b?	Real*8	(NumObs)
FreqRateIon	Effective Group Rate Ionospheric Fre-quency	Obs	ObsDerived	EffFreq_b?	Real*8	(NumObs)
FRNGERR	Fourfit error flag blank=OK.	Obs	Observables	CorrInfo_b?	Char*2	(NumObs)
FRQGROUP	Frequency group code.	Obs	Observables	CorrInfo_b?	Char*2	(NumObs)
GammaPart	Consensus partial w.r.t. Gamma	Obs	ObsPart	Part-Gamma	Real*8	(TimeDim2, NumObs)
GeocMBD	Tot geocenter group delay (sec).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
GeocPhase	Tot phase ref to cen of Earth.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
GeocRate	Tot geocenter delay rate (s/s).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
GeocResidPhase	Resid phs corrected to cen of E.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
GeocSBD	Tot geocenter sbd delay (sec).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
GradOffsetConstraint	Gradient Offset Constraint	Session	Solve	AtmSetup	Real*8	(NumStat)
GradRateConstraint	Gradient Rate Constraint	Session	Solve	AtmSetup	Real*8	(NumStat)
GroupBLWeightName	B.L.names for formal errors	Session	Session	GroupBLWeights	Char*8	(2, XXX)
GroupBLWeights	Group delay and rate re-weighting con-stants.	Session	Session	GroupBLWeights	Real*8	(XXX, 2)
GroupDelay	Observed group delay in sec.	Obs	OBSOLETE	-	Real*8	(NumObs)
GroupDelayFull	Delay Observable with ambiguities re-solved and added.	Obs	ObsEdit	GroupDelayFull_b?	Real*8	(NumObs)
GroupDelaySig	Delay Measurement Sigma	Obs	Observables	GroupDelay_b?	Real*8	(NumObs)
GroupRate	Rate Observable	Obs	Observables	GroupRate_b?	Real*8	(NumObs)

GroupRateSig	Rate Measurement Sigma	Obs	Observables	GroupRate_b?	Real*8	(NumObs)
History	Description of cable-cal processing	Station	-	Cal-Cable	Char*60	-
IDELAY	Corel instrumental delay (sec).	Obs	Observables	CorrInfo_b?	Real*8	(2, NumObs)
IERSDES	IERZS Designation array	Session	Apriori	StationApriori	Char*8	-
INCOH2	Incoh amp from FRNGE plot segs.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
INCOHAMP	Fr. amp from incoh int of chan.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
INDEXNUM	Corel index numbers by sb freq.	Obs	Observables	CorrInfo_b?	Integer*4	(2,16,NumObs)
IonBits	ICORR for full ion tracking.	Obs	Solve	IonBits	Integer*4	(NumObs)
IonGroupCal	Ion correction. Add to theo. sec	Obs	ObsDerived	Cal-IonGroup_b?	Real*8	(TimeDim2,NumObs)
IonGroupCalDataFlag	0=OK, -1=Missing, -2=bad	Obs	ObsDerived	Cal-IonGroup_b?	Integer*4	(NumObs)
IonGroupCalSigma	Ion correction to sigma. sec	Obs	ObsDerived	Cal-IonGroup_b?	Real*8	(TimeDim2,NumObs)
IonPhaseCal	Phase Corr Iono	Obs	ObsDerived	Cal-IonPhase_b?	Real*8	(NumObs)
IonPhaseCalSigma	Phase Corr RMS	Obs	ObsDerived	Cal-IonPhase_b?	Real*8	(NumObs)
IonSolveFlag	Bit flag indicating station has iono correction	Session	Solve	IonSetup	Integer*4	(NumStation)
IonStations	Stations with ionocorrection	Session	Solve	IonSetup	Char*8	(NumStation)
iUTCInterval	First and last UTC time tag in input file.	Session	-	Head	Integer*4	(5,2)
LeapSecond	Leap second	Session	Solve	Misc	Real*8	-
LOFreq	LO frequencies per cha/sta MHz.	Obs	Observables	CorrInfo_b?	Real*8	(2,NumChannels,NumObs)
MJD	MJD time tag of scan	Scan	Solve	ScanTimeMJD	Integer*4	(NumScan)
MoonXYZ	Lunar geocentric coordinates.	Scan	Scan	Ephemeris_kDE405JPL	Real*8	(3,TimeDim2, NumScan)
NGSQualityFlag	Zero means good. A Combination of DelayFlag and IonCode	Obs	ObsEdit	NGSQualityFlag	Integer*4	(NumObs)
NMFDryCal	Nhmf (dry) atm. contribution	Station	-	Cal-NMFDry	Real*8	(TimeDim2, NumStatScan)
NMFDryPart	Nhmf2 dry partial deriv. def.	Station	-	Part-NMFDry	Real*8	(TimeDim2, NumObs)

NMFGradEWPart	NMF dry atm. EW Gradient Partials	Obs	ObsPart	Part-NMFGrad	Real*8	(TimeDim2, NumObs)
NMFGradNSPart	NMF dry atm. NS Gradient Partials	Obs	ObsPart	Part-NMFGrad	Real*8	(TimeDim2, NumObs)
NMFGradPart	Niell dry atm. gradient partials	Obs	-	NGRADPAR	Real*8	(2,2,2)
NMFWetCal	Whmf (wet) atm. contribution	Station	-	Cal-NMFWet	Real*8	(TimeDim2, Num-StatScan)
NMFWetPart	NMF Wet Partial	Station	-	Part-NMFWet	Real*8	(TimeDim2, NumObs)
NumAccum	No. of accum. periods in Channel	Obs	Observables	CorrInfo_b?	Inte-ger*4	(32, NumObs)
NumAp	# of AP by sideband and channel.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(2,NumChannels,NumObs)
NumBBCFreqs	No. of BBC frequencies in Band	Obs	Observables	CorrInfo_b?	Inte-ger*4	(NumObs)
NumChannels	No. of U-L pairs in integration.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(NumObs)
NumClockRefSite	# of clock reference stations	Session	Solve	ClockSetup	Inte-ger*4	-
NumGroupAmbig	Number of group delay ambiguities	Obs	ObsEdit	NumGroupAmbig_b?	Inte-ger*4	(NumObs)
NumLagsUsed	Num of lags used for correlation	Session	Solve	Misc	Inte-ger*4	-
NumObs	Number of observations (I*4)	Session	-	Head	Inte-ger*4	-
NumPhaseAmbig	Number of phase ambiguities	Obs	ObsEdit	NumPhaseAmbig_b?	Inte-ger*4	(NumObs)
NumSamples	# of samples by sideband and cha	Obs	Observables	CorrInfo_b?	Real*8	(2,NumChannels,NumObs)
NumScan	Number of Scans (Integer*4)	Session	-	Head	Inte-ger*4	-
NumScansPerStation	Number of scans per station.	Session	CrossRefer-ence	StationCrossRef	Inte-ger*4	(NumStation)
NumSource	Number of radio sources.	Session	-	Head	Inte-ger*4	-
NumStation	Number of sites.	Session	-	Head	Inte-ger*4	-
Nut2000PsiEps	IAU200A Nut. - Dpsi Deps Rates	Scan	Scan	Nut_k2000PsiEps	Real*8	(2,TimeDim2, Num-

						Scan)
Nut2000PsiEpsPart	IAU2000A Nutation Psi Eps Partls	Obs	ObsPart	Part-Nut2KPsiEps	Real*8	(2,TimeDim2, NumObs)
Nut2000XYS	CIP Coordinates X Y S and Rates	Scan	Scan	Nut_k2000XYS	Real*8	(3, TimeDim2, Num-Scan)
Nut2KXYPart	IAU2000A Nutation X Y Partials	Obs	ObsPart	Part-Nut2KXY	Real*8	(2,TimeDim2, NumObs)
NutationFlag	Nutation flow control mess def.	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
NutationMessage	Nutation message definition	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
NutPsiEpsPart	Pre Calc10 Nuation Partial	Obs	ObsPart	Part-NutPsiEps	Real*8	(2,TimeDim2, NumObs)
NutWahrCal	2000A Nut to Wahr Nut Contributn	Obs	ObsCalTheo	Cal-NutWahr	Real*8	(TimeDim2,NumObs)
NutWahrPsiEps	Wahr nut vals - Dpsi Deps&rates	Scan	Scan	Nut_kWahr	Real*8	(2,TimeDim2, Num-Scan)
Obs2Baseline	Cross reference from observation to baseline	Obs	CrossRefer-ence	ObsCrossRef	Inte-ger*4	(2,NumObs)
Obs2Scan	Cross reference from observation to scan	Obs	CrossRefer-ence	ObsCrossRef	Inte-ger*4	(NumObs)
ObsCalFlag	Bit set indicate that calibration is rec-ommended.	Session	Solve	CalibrationSetup	Inte-ger*4	-
ObsCalName	Available obs dependent calibrations (poletide, earthdide, …)	Session	Solve	CalibrationSetup	Char*	(14)
OCCUPNUM	Site Occupation Number.	Obs	Observables	CorrInfo_b?	Char*8	(2,NumObs)
OceanAmpEW	Ocean amp. E-W component(meters)	Session	ObsTheoreti-cal	CalcInfo	Real*8	(NumStation,2)
OceanAmpNS	Ocean amp. N-S component(meters)	Session	ObsTheoreti-cal	CalcInfo	Real*8	(NumStation,2)
OceanAmpUp	Ocean amp. radial comp.-meters.	Session	ObsTheoreti-cal	CalcInfo	Real*8	(NumStation,2)
OceanFlag	Ocean load flow control mess def	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
OceanHorizontalAmp	Horz ocean loading ampltudes (m)	Session	ObsTheoreti-cal	CalcInfo	Real*8	(11,2,NumStation)
OceanHorizontal-Phase	Horz ocean loading phases (rad).	Session	ObsTheoreti-cal	CalcInfo	Real*8	(11,2,NumStation)
OceanLoadCal	Obs dependent ocean loading	Obs	ObsCalTheo	Cal-OceanLoad	Real*8	(TimeDim2,NumStatSca

						n)
OceanLoadDis	Ocean load site depndnt displace	Station	-	Dis_kOceanLoad	Real*8	(3, TimeDim2,NumStatScan)
OceanLoadHorizCal	Site-dep ocean cont - horizontal	Station	-	Cal-OceanLoad	Real*8	(TimeDim2, Num-StatScan)
OceanLoadVertCal	Site-dep ocean cont - vertical	Station	-	Cal-OceanLoad	Real*8	(TimeDim2, Num-StatScan)
OceanMessage	Ocean loading message definition	Session	ObsTheoretical	CalcInfo	Char*80	-
OceanPhaseEW	Ocean phase E-W component (rad)	Session	ObsTheoretical	CalcInfo	Real*8	(NumStation,2)
OceanPhaseNS	Ocean phase N-S component (rad)	Session	ObsTheoretical	CalcInfo	Real*8	(NumStation,2)
OceanPhaseUp	Ocean phase radial comp.-radians	Session	ObsTheoretical	CalcInfo	Real*8	(NumStation,2)
OceanStations	Ocean loading station status.	Session	ObsTheoretical	CalcInfo	Char*4	(NumStation)
OceanUpAmp	Vert ocean loading ampltudes (m)	Session	ObsTheoretical	CalcInfo	Real*8	(NumStation,11)
OceanUpPhase	Vert ocean loading phases (rad).	Session	ObsTheoretical	CalcInfo	Real*8	(NumStation,11)
ORIGFILE	Original COREL file name.	Obs	Observables	CorrInfo_b?	Char*6	(NumObs)
ParallaxCal	Parallax partial/contr 1 parsec	Obs	ObsCalTheo	Cal-Parallax	Real*8	(TimeDim2,NumObs)
ParallaxFlag	Parallax flow control mess def	Session	ObsTheoretical	CalcInfo	Char*80	-
ParallaxMessage	Parallax message definition	Session	ObsTheoretical	CalcInfo	Char*80	-
ParallaxPart	Parallax partial deriv. def.	Obs	ObsPart	Part-Parallax	Real*8	(TimeDim2, NumObs)
PepMessage	PEP Utility Message Definition	Session	ObsTheoretical	CalcInfo	Char*80	-
Phase	Total phase	Obs	Observables	Phase_b?	Real*8	(NumObs)
PhaseBLWeights	Phase formal error constants	Session	Solve	PhaseBLWeights	Real*8	(2, NumBaselines)
PHASECAL	PC rate by sta (us per s).	Obs	Observables	CorrInfo_b?	Real*8	(2, NumObs)
PhaseCalAmpPhase	PC amp phs frq by sta channel.	Obs	Observables	CorrInfo_b?	Inte-	(3,2,NumChannels,Num

					ger*4	Obs)
PhaseCalOffset	Phase cal offset (-18000/18000).	Obs	Observables	CorrInfo_b?	Inte-ger*4	(2,NumChannels, NumObs)
PhaseDataFlag	0=OK, -1=Missing, -2=bad,-3=sigma small, -4=sigma big	Obs	Observables	Phase_b?	Inte-ger*4	(NumObs)
PhaseDelay	Phase delay resolved with ambiguities added.	Obs	ObsEdit	PhaseDelayFull_b?	Real*8	(NumObs)
PhaseDelayFull	Phase delay resolved with ambiguities added.	Obs	ObsEdit	PhaseDelayFull_b?	Real*8	(NumObs)
PhaseDelayRaw	Unresolved phase delay	Obs	Observables	Phase_b?	Real*8	(NumObs)
PhaseFlag	Phase unweight flag	Obs	ObsEdit	Edit	Inte-ger*4	(NumObs)
PhaseSig	Phase delay sigma.	Obs	Observables	Phase_b?	Real*8	(NumObs)
Polarization	Polarization per sta/chan R/L.	Obs	Observables	CorrInfo_b?	Char*	(NumChan-nels,NumObs)
PolarMotion	Polar motion X & Y for obs (rad)	Scan	Scan	EOPApriori	Real*8	(2, NumScan)
PoleTideCal	Pole tide contributions def.	Obs	ObsCalTheo	Cal-PoleTide	Real*8	(TimeDim2,NumObs)
PoleTideFlag	Pole tide flow control mess def	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
PoleTideMessage	Pole tide message definition	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
PoleTideOldRestore-Cal	Old Pole Tide Restorer Contrib.	Obs	ObsCalTheo	Cal-PoleTideOldRestore	Real*8	(TimeDim2,NumObs)
PrecessionData	Precession constant (asec/cent).	Session	ObsTheoreti-cal	CalcInfo	Real*8	-
PrecessionFlag	Precession flow contril mess def	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
PrecessionMessage	Precession message definition	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
PrecessionPart	Precession partial deriv. def.	Obs	ObsPart	Part-Precession	Real*8	(TimeDim2, NumObs)
PrincipalInvestigator	Agency/contact_person/PI name.	Session	-	Head	Char*80	-
ProbFalseDetection	Prob of false det from FRNGE.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
QBFACTOR	Measure of uniformity of data.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
QualityCode	FRNGE quality index 0 --> 9	Obs	Observables	QualityCode_b?	Char*1	(NumObs)

Commented [DV4]: TimeDim2?

RaDecPart	Star partial derivatives def.	Obs	ObsPart	Part-RaDec	Real*8	(2,TimeDim2, NumObs)
RateFlag	Delay rate unweight flag.	Obs	ObsEdit	Edit	Integer*4	(NumObs)
RateTheoretical	Consensus theoretical rate	Obs	ObsTheoretical	RateTheoretical	Real*8	(NumObs)
RATOBSVM	Obs rate at central epoch .	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
RATRESID	Rate resid (sec per sec).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
RecordingMode	Recoding mode.	Session	-	Head	Char*80	-
RECSETUP	Samp rate(kHz) Frames/PP PP/AP.	Obs	Observables	CorrInfo_b?	Integer*4	(3,NumObs)
RECTRACK	Trk table by sta sideb channel.	Obs	Observables	CorrInfo_b?	Integer*4	(2,2,14,NumObs)
REFCLKER	Ref sta clock epoch microsec.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
ReferenceClock	Batchmode clock reference site	Session	Solve	ClockSetup	Char*8	-
RefFreq	Frequency to which phase is referenced	Obs	Observables	RefFreq_b?	Real*8	-
RelativityFlag	Relativisitc bending use status	Session	ObsTheoretical	CalcInfo	Char*60	-
RelativityMessage	Relativity mod data (gamma).	Session	ObsTheoretical	CalcInfo	Real*8	-
RelHum	Rel.Hum. at local WX st (50%=.5)	Station	-	Met	Real*8	(NumStatScan)
RunCode	Run code e.g. "329-1300".	Obs	Observables	CorrInfo_b?	Char*8	(NumObs)
S2EFFREQ	Effective group freq for ion.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
S2PHEFRQ	Effective phase frequency	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
S2REFREQ	Effective frequency for rate	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
SampleRate	Sample rate (Hz).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
SBDelay	Single band delay	Obs	Observables	SBDelay_b?	Real*8	(NumObs)
SBDelaySig	Single band delay error	Obs	Observables	SBDelay_b?	Real*8	(NumObs)
SBRESID	Single band delay residual.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
Scan2Source	Cross reference scan to source	Session	CrossReference	SourceCrossRef	Integer*4	(NumScan)
Scan2Station	Cross reference scans to station	Session	CrossReference	StationCrossRef	Integer*4	(NumStation, NumScan)
ScanName	Scanname in database	Scan	Scan	ScanName	Char*10	(NumScan)

ScanNameFull	ScanName=UTC TimeTag + SourceName	Scan	Scan	ScanName	Char*30	(NumScan)
Second	Seconds part of time tag for scan	Scan	Scan	TimeUTC	Real*8	(NumScan)
SiteMessage	Site Module Message Definition	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
SiteZenithDelay	Site zenith path delays (nsec).	Session	ObsTheoreti-cal	CalcInfo	Real*8	(NumStation)
SNR	Signal to noise ratio for S-band	Obs	Observables	SNR_b?	Real*8	(NumObs)
Source	Radio source name.	Obs	Observables	Source	Char*8	(NumObs)
Source2000RaDec	J2000 Source RA and Dec	Session	Apriori	SourceApriori	Real*8	(2,NumSource)
SourceList	Source names array.	Session	-	Head	Char*8	(NumSource)
SourceNameApriori	Source names in RA order.	Session	Apriori	SourceApriori	Char*8	(NumSource)
SourceNameCrossRef	Source names.	Session	CrossRefer-ence	SourceCrossRef	Char*8	(NumSource)
SourceReference	Source of coordinate values.	Session	Apriori	SourceApriori	Char*20	(NumSource)
SourceSelectionFlag	Source selection status bit-mapped ar-ray.	Session	Solve	SelectionStatus	Inte-ger*4	(70)
SRCHPAR	FRNGE/Fourfit search parameters.	Obs	Observables	CorrInfo_b?	Real*8	(6,NumObs)
STARELEV	Elev angles calc by COREL.	Obs	Observables	CorrInfo_b?	Real*8	(2,NumObs)
StarMessage	Star module message definition	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
StarParallaxFlag	Parallax flow control mess def	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
StartOffset	Offset nominal start time (sec).	Obs	Observables	CorrInfo_b?	Inte-ger*4	(NumObs)
STARTSEC	Start time in sec past hour.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
StatCalFlag	Bit set indicate that calibration is rec-ommended.	Session	Solve	CalibrationSetup	Inte-ger*4	(NumStation)
StatCalName	Station depedendent calibratipns (Ca-ble, Phase, etc…)	Session	Solve	CalibrationSetup	Char*8	(6)
Station2Scan	Cross reference station-scan to sched-ule-scan	Session	CrossRefer-ence	StationCrossRef	Inte-ger*4	(NumStation, NumScan)
StationList	Site names array.	Session	-	Head	Char*8	(NumStat)
StationNameApriori	Site names in alphabetical order	Session	Apriori	StationApriori	Char*8	(NumStation)

StationNameCable	Stations for cable sign	Session	Solve	Misc	Char*8	(NumStation)
StationNameCrossRef	Site names in alphabetical order	Session	CrossRefer-ence	StationCrossRef	Char*	(NumStation)
StationXYZ	Site cartesian coords (m).	Session	Apriori	StationApriori	Real*8	(3, NumStation)
StopOffset	Offset nominal stop time (sec).	Obs	Observables	CorrInfo_b?	Inte-ger*4	(NumObs)
StopSec	Stop time in sec past hour.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
SunXYZ	Solar geocentric coordinates.	Scan	Scan	Ephemer-is_kDE405JPL	Real*8	(3, TimeDim2, Num-Scan)
TapeCode	Tape quality code.	Obs	Observables	CorrInfo_b?	Char*6	(NumObs)
TAPEID	Raw data tape ID for ref and rem	Obs	Observables	CorrInfo_b?	Char*8	(2,NumObs)
TempC	Temp in C at local WX station	Station	-	Met	Real*8	(NumStatScan)
TheoryMessage	Theory module identification	Session	ObsTheoreti-cal	CalcInfo	Char*80	-
TidalUT1Flag	Flag for tidal terms in UT1 sers	Session	ObsTheoreti-cal	CalcInfo	Inte-ger*4	-
TiltRemoverCal	Axis Tilt Contribution Remover	Obs	ObsCalTheo	Cal-TiltRemover	Real*8	(TimeDim2,NumObs)
TimeSinceStart	Interval since start time (sec).	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
TotalFringeErr	Total fringe phase error (deg)	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
TOTPCENT	Tot phase at central epoch.	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
Tsys	TSYS TMP (K) 14CH AND IF1 IF2	Obs	Observables	Tsys	Real*8	(16,2)
UNPHAFLG	UnPhaseCal flag	Obs	Solve	UnPhaseCal	Inte-ger*4	(2,2)
UnPhaseCal	UnPhaseCal effect - group&rate	Obs	ObsCalTheo	Cal-UnphaseCal	Real*8	(2, TimeDim2,NumObs)
URVR	Rate derivatives mHz per asec.	Obs	Observables	CorrInfo_b?	Real*8	(2,NumObs)
UserSup	User action for suppression	Obs	ObsEdit	Edit	Inte-ger*4	(NumObs)
UT1	UT1 time of day for this obsvr.	Scan	Scan	EOPApriori	Real*8	(NumScan)
UT1ArrayInfo	Array: (FJD of start, spacing in days, number points, Scaling (should be 1))	Session	Solve	CalcEop	Real*8	(4)
UT1EPOCH	TAI - UT1 epoch value definition	Session	ObsTheoreti-cal	CalcInfo	Real*8	(2,2)
UT1FLag	UT1 control flag message def.	Session	ObsTheoreti-	CalcInfo	Char*80	-

			cal			
UT1IntrpMode	Message for UT1 interp. scheme	Session	Solve	CalcEop	Char*60	-
UT1OffsetConstraint	UT1 Offset Constraint	Session	Solve	EopSetup	Real*8	-
UT1Origin	Final Value TAI-UT1 origin text.	Session	Solve	CalcEop	Char*80	-
UT1OrthoCal	ORTHO_EOP Tidal UT1 contribution	Obs	ObsCalTheo	Cal-HFEOP-IERS2003	Real*8	(TimeDim2,NumObs)
UT1Part	UT1 partial derivatives def.	Obs	ObsPart	Part-EOP	Real*8	(2,TimeDim2, NumObs)
UT1Values	Final Value TAI-UT1 data points.	Session	Solve	CalcEop	Real*8	(15)
UTCCorr	UTC time tag of correlation.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(6,NumObs)
UTCErr	A priori UTC error site 1 (sec)	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
UTCMidObs	UTC at central epoch YMDHMS.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(6,NumObs)
UTCProc	YDDD of COREL by sta channel.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(2,14,NumObs)
UTCScan	Nominal scan time YMDHMS.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(6,NumObs)
UTCVLB2	UTC of FRNGE proc YMDHMS.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(6,NumObs)
UVFperAsec	U V in FR per arcsec from CALC	Obs	Solve	UVFperAsec	Real*8	(2, NumObs)
VFDWELL	Dwell time in each channel (sec)	Obs	Observables	CorrInfo_b?	Real*8	(NumObs)
VFRQAM	Normalized channel amplitude	Obs	Observables	CorrInfo_b?	Real*8	(32,NumObs)
VFRQPCAM	Phase cal tone Amplitudes	Obs	Observables	CorrInfo_b?	Real*8	(2,32,NumObs)
VFRQPCFR	Phase cal tone Frequencies	Obs	Observables	CorrInfo_b?	Real*8	(2,32,NumObs)
VFRQPCPH	PHASE cal tone Phases	Obs	Observables	CorrInfo_b?	Real*8	(2,32,NumObs)
VFRQPH	Channel Phase (degrees)	Obs	Observables	CorrInfo_b?	Real*8	(32,NumObs)
VIRTFREQ	Sky Frequencies	Obs	Observables	CorrInfo_b?	Real*8	(32,NumObs)
VLB1FILE	Correlator file name.	Obs	Observables	CorrInfo_b?	Char*6	(NumObs)
VLB1XTNT	corr. ext by sideb channel.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(NumObs)
VLB2PRG	FRNGE(YYMMDD) Fourfit(x.x) ver.	Obs	Observables	CorrInfo_b?	Char*6	(NumObs)
VLB2XTNT	FRNGE extent number.	Obs	Observables	CorrInfo_b?	Inte-ger*4	(NumObs)

WobArrayInfo	Array: (FJD of start, spacing in days, number points)	Session	Solve	CalcEop	Real*8	(3)
WobbleFlag	Wobble flow control mess def.	Session	ObsTheoretical	CalcInfo	Char*80	-
WobbleOrigin	Final Value wobble origin text.	Session	Solve	CalcEop	Char*80	-
WobblePart	Wobble partial derivatives def.	Obs	ObsPart	Part-EOP	Real*8	(2,TimeDim2, NumObs)
WOBEPOCH	Interpolated wobble array def	Session	ObsTheoretical	CalcInfo	Real*8	(2,2)
WobIntrpMode	Interp. scheme for polar motion.	Session	Solve	CalcEop	Char*60	-
WobNutCal	Short period nutation wobble con	Obs	ObsCalTheo	Cal-WobNut	Real*8	(TimeDim2,NumObs)
WobOffsetConstraint	Polar Motion Offset Constraint	Session	Solve	EopSetup	Real*8	-
WobOrthoCal	ORTHO_EOP tidal wobble contribtn	Obs	ObsCalTheo	Cal-HFEOP-IERS2003	Real*8	(TimeDim2,NumObs)
WobPart	Pole Tide Partials w.r.t. X & Y	Obs	ObsPart	Part-Poletide	Real*8	(2,TimeDim2, NumObs)
WobValues	Final wobble X Y component value	Session	Solve	CalcEop	Real*8	(2,15)
WVRDelay	WVR LINE OF SIGHT & ZENITH(NSEC)	Station	-	WVR	Real*8	(2,2,2)
WVRQcode	WVR DELAY DATA QUALITY CODE	Station	-	WVR	Integer*4	(2)
WVRTemp	BRT TEMP (K) OF WVR ,F1,F2.	Station	-	WVR	Real*8	(2,2)
WVRTempRef	REFERENCE1,2 WVR TEMP (K).	Station	-	WVR	Real*8	(3,2)
XwobbleCal	X Wobble contribution definition	Obs	ObsCalTheo	Cal-Wobble	Real*8	(TimeDim2,NumObs)
XYZPart	Site partials: dtau/dr_1=-dtau/dr_2	OBs	ObsPart	Part-XYZ	Real*8	(3,TimeDim2, NumObs)
YMDHM	YMDHM time tag of scan	Scan	Scan	TimeUTC	Integer*4	(5, NumScan)
YwobbleCal	Y Wobble contribution definition	Obs	ObsCalTheo	Cal-Wobble	Real*8	(TimeDim2,NumObs)
ZDELAY	Corel zenith atmos. delay (sec).	Obs	Observables	CorrInfo_b?	Real*8	(2,NumObs)

D Mapping of Lcodes onto vgosDB variables

This appendix indicates how Lcodes are mapped to vgosDB variables. If the directory is given as OBSOLETE the LCODE is not mapped into a vgosDB variable.

LCODE	vgosDB_variable_name	Description	Directory	File
# AMBIG	NumGroupAmbig	Number of group delay ambiguities	ObsEdit	NumGroupAmbig_bX
# CLK_RF	NumClockRefSite	# of clock reference stations	Solve	ClockSetup
# DELAYS	# DELAYS	NO. of delays in this obs .	OBSOLETE	-
# RATES	# RATES	NO. of rates in this obs .	OBSOLETE	-
# SITES	NumStation	Number of sites.	-	Head
# STARS	NumSource	Number of radio sources.	-	Head
#CHANELS	NumChannels	No. of U-L pairs in integration.	Observables	CorrInfo_bX
#GAMBG_S	NumGroupAmbig	Number of group delay ambiguities	ObsEdit	NumGroupAmbig_bS
#PAMBG_S	NumPhaseAmbig	Number of phase ambiguities	ObsEdit	NumPhaseAmbig_bS
#PAMBIG	NumPhaseAmbig	Number of phase ambiguities	ObsEdit	NumPhaseAmbig_bX
#SAMPLES	NumSamples	# of samples by sideband and cha	Observables	CorrInfo_bX
#VFREQS	NumBBCFreqs	No. of BBC frequencies in Band	Observables	CorrInfo_bX
A1 - TAI	A1minusTAI	TAI USNO- TAI BIH = 0.03439 sec.	OBSOLETE	-
A1 - UTC	A1minusUTC	FJD A1-UTC (sec) rate off.(sec)	OBSOLETE	-
ABASACCE	ABASACCE	Corel bas/apr accel (1/sec**2).	Observables	CorrInfo_bX
ABASDEL	ABASDEL	Corel bas/apr delay (sec).	Observables	CorrInfo_bX
ABASRATE	ABASRATE	Corel bas/apr delay rate (s/s).	Observables	CorrInfo_bX
AC_SITES	AtmRateSite	Site list for clocks constraints	Solve	AtmSetup
ACCELGRV	ACCELGRV	Accel grav at erth surface m/s^2	OBSOLETE	-
AMPBYFRQ	ChanAmpPhase	Amp(0-1) phs by chan(-180to180)	Observables	CorrInfo_bX
APCLOFST	APCLOFST	Apriori clock offset microsec.	Observables	CorrInfo_bX
APLENGTH	Aplength	Length of accumul. period in sec	Solve	Misc
ARCHIVE	ARCHIVE	Microfiche, B tape number.	OBSOLETE	-
ATI CFLG	ATIFlag	ATIME Flow Control Message Def.	ObsTheoretical	CalcInfo

ATI MESS	ATIMessage	ATIME Message Definition	ObsTheoretical	CalcInfo
ATM CFLG	ATMFlag	Atmosphere control flag mess def	ObsTheoretical	CalcInfo
ATM FL 1	ATM FL 1	List of atmosphere flags	OBSOLETE	-
ATM FL 2	ATM FL 2	List of atmosphere flags	OBSOLETE	-
ATM FL 3	ATM FL 3	List of atmosphere flags	OBSOLETE	-
ATM FL 4	ATM FL 4	List of atmosphere flags	OBSOLETE	-
ATM FL 5	ATM FL 5	List of atmosphere flags	OBSOLETE	-
ATM FL 6	ATM FL 6	List of atmosphere flags	OBSOLETE	-
ATM FL 7	ATM FL 7	List of atmosphere flags	OBSOLETE	-
ATM FL 8	ATM FL 8	List of atmosphere flags	OBSOLETE	-
ATM MESS	ATMMessage	Atmosphere message definition	ObsTheoretical	CalcInfo
ATM PRES	AtmPres	Pressure in hPa at site	-	Met
ATM_CFLG	#N/A	#N/A	#N/A	#N/A
ATM_CNST	AtmRateConstraint	Atmosphere constraint. ps/hr	Solve	AtmSetup
ATM_INTV	AtmInterval	Batchmode atmos interval - hours	Solve	AtmSetup
ATMEPO 1	ATMEPO 1	Atmosphere epochs - J.D.s	OBSOLETE	-
ATMEPO 2	ATMEPO 2	Atmosphere epochs - J.D.s	OBSOLETE	-
ATMEPO 3	ATMEPO 3	Atmosphere epochs - J.D.s	OBSOLETE	-
ATMEPO 4	ATMEPO 4	Atmosphere epochs - J.D.s	OBSOLETE	-
ATMEPO 5	ATMEPO 5	Atmosphere epochs - J.D.s	OBSOLETE	-
ATMEPO 6	ATMEPO 6	Atmosphere epochs - J.D.s	OBSOLETE	-
ATMEPO 7	ATMEPO 7	Atmosphere epochs - J.D.s	OBSOLETE	-
ATMEPO 8	ATMEPO 8	Atmosphere epochs - J.D.s	OBSOLETE	-
AUTO_SUP	AUTO_SUP	Automatic suppression status	OBSOLETE	-
AUTOEDIT	AUTOEDIT	1=Run resulted from AUTOEDIT.	Observables	CorrInfo_bX
AXISOFFS	AxisOffset	Axis offsets (m).	Apriori	AntennaApriori
AXISTILT	AxisTilt	Fixed axis tilt	Apriori	AntennaApriori
AXISTYPS	AxisType	Axis type (1-eq 2-xy 3-azel 4 5)	Apriori	AntennaApriori
AXO CFLG	AxisOffsetFlag	Axis Offset Control flag mes def	ObsTheoretical	CalcInfo
AXO CONT	AxisOffsetCal	New Axis Offset Contributions	-	Cal-AxisOffset

AXO MESS	AxisOffsetMessage	Axis Offset Message Definition	ObsTheoretical	CalcInfo
AXO PART	AxisOffsetPart	Axis Offset partial deriv. def.	-	Part-AxisOffset
AZ-THEO	AzTheo	Azimuth array definition	-	AzEl
BAROCALS	BAROCALS	Baro offset (mbar) and Ht. (m)	OBSOLETE	-
BARONAME	BARONAME	Names for barometer calibrations	OBSOLETE	-
BASCLCKS	BaselineClock	List of baseline dependent clocks	Solve	BaselineClockSetup
BASELINE	Baseline	Ref and rem site names.	Observables	Baseline
BASLSTAT	BaselineSelectionFlag	Baseline selection bit maped array. 1=some obs, etc.	Solve	SelectionStatus
BATCHCNT	BATCHCNT	List of baseline dependent clocks	OBSOLETE	-
BBC IND	BBCIndex	Physical BBC number by channel.	Observables	CorrInfo_bX
BENDPART	BendPart	Grav. bend. partial w.r.t. Gamma	ObsPart	Part-Bend
BITSAMPL	BITSAMPL	Number of bits per sample.	Observables	CorrInfo_bX
BLDEPCKS	BaselineClock	Bl-dependent clock list	Solve	BaselineClockSetup
BRK_EPOC	ClockBreakEpoch	Batchmode clock break epochs	Session	ClockBreak
BRK_FLAG	ClockBreakFlag	Batchmode clock break flags	Session	ClockBreak
BRK_NUMB	BRK_NUMB	Number of batchmode clock breaks	Session	ClockBreak
BRK_SITE	ClockBreakSite	Sites with real clock breaks	Session	ClockBreak
BRK_SNAM	ClockBreakSite	Batchmode clock break stations	Session	ClockBreak
BRK_STAT	ClockBreakSite	Batchmode clock break stations	Session	ClockBreak
BRT TEMP	WVRTemp	BRT TEMP (K) OF WVR ,F1,F2.	-	WVR
CABL DEL	CableCal	Cable calibration data	-	Cal-Cable
CAL FLGS	StatCalFlag	Bit set indicate that calibration is recom-mended.	Solve	CalibrationSetup
CAL LIST	StatCalName	Station depedendent calibratipns (Cable, Phase, etc…)	Solve	CalibrationSetup
CALBYFRQ	PhaseCalAmpPhase	PC amp phs frq by sta channel.	Observables	CorrInfo_bX
CALC VER	CalcVersion	CALC version number	ObsTheoretical	CalcInfo
CALCFLGN	CalcFlagNames	CALC flow control flags name def	ObsTheoretical	CalcInfo
CALCFLGV	CalcFlagValues	CALC flow control flags valu def	ObsTheoretical	CalcInfo

CALSITES	CalStationName	List of sites for standard cal	Solve	CalibrationSetup
CBL SIGN	CableSign	Signs of cable cal application	Solve	Misc
CBL STAT	StationNameCable	Stations for cable sign	Solve	Misc
CC_SITES	ClockRateName	Site list for clocks constraints	Solve	ClockSetup
CCOR ERR	CorCofErr	Corr. Coeff. formal error	Observables	CorrInfo_bX
CF2J2K	CF2J2K	Crust-fixed to J2000 Rot. Matrix and de-riviatives	Scan	Rot_kCF2J2K
CHAN ID	ChannelID	One-letter Fourfit channel ID.	Observables	CorrInfo_bX
CLCKEP 3	CLCKEP 3	Clock epochs (JD,fraction)	OBSOLETE	-
CLCKEP 4	CLCKEP 4	Clock epochs (JD,fraction)	OBSOLETE	-
CLCKEP 5	CLCKEP 5	Clock epochs (JD,fraction)	OBSOLETE	-
CLCKEP 6	CLCKEP 6	Clock epochs (JD,fraction)	OBSOLETE	-
CLCKEP 7	CLCKEP 7	Clock epochs (JD,fraction)	OBSOLETE	-
CLCKEP 8	CLCKEP 8	Clock epochs (JD,fraction)	OBSOLETE	-
CLCKFL 3	CLCKFL 3	List of clock flags	OBSOLETE	-
CLCKFL 4	CLCKFL 4	List of clock flags	OBSOLETE	-
CLCKFL 5	CLCKFL 5	List of clock flags	OBSOLETE	-
CLCKFL 6	CLCKFL 6	List of clock flags	OBSOLETE	-
CLCKFL 7	CLCKFL 7	List of clock flags	OBSOLETE	-
CLCKFL 8	CLCKFL 8	List of clock flags	OBSOLETE	-
CLK_CFLG	CLK_CFLG	Clock constraint use flag.	Solve	ClockSetup
CLK_CNST	ClockRateConstraint	Clock constraint-Parts in 1.e14	Solve	ClockSetup
CLK_INTV	ClockInterval	Batchmode clock interval - hours	Solve	ClockSetup
CLK_SITE	ReferenceClock	Batchmode clock reference site	Solve	ClockSetup
CLK_SITS	ReferenceClock	List of clock reference stations	Solve	ClockSetup
CLKBREAK	CLKBREAK	Status of clock break existence	Session	ClockBreak
CLODRACM	ClockAprioriRate	A priori clock drift (sec/sec)	Apriori	ClockApriori
CLOOFACM	ClockAprioriOffset	A priori clock offset (sec)	Apriori	ClockApriori
COHCOR_S	Correlation	Corr coeff (0 --> 1) for S-band	Observables	Correlation_bS
COHERCOR	Correlation	Corr coeff (0 --> 1).	Observables	Correlation_bX

COMMENT2	COMMENT2	FRNGE runtime comments.	OBSOLETE	-
CON CONT	BendCal	Consensus bending contrib. (sec)	ObsCalTheo	Cal-Bend
CONSNDEL	DelayTheoretical	Consensus theoretical delay	ObsTheoretical	DelayTheoretical
CONSNRAT	RateTheoretical	Consensus theoretical rate	ObsTheoretical	RateTheoretical
CONSPART	GammaPart	Consensus partial w.r.t. Gamma	ObsPart	Part-Gamma
COR DATA	CoronaData	Corona model parameters.	ObsTheoretical	CalcInfo
CORBASCD	CORBASCD	Correlator baseline code (2 ch).	Observables	CorrInfo_bX
CORBASNO	CORBASNO	Correlator baseline number.	Observables	CorrInfo_bX
CORCLOCK	CORCLOCK	Clock offset(sec)/rate(sec/sec).	Observables	CorrInfo_bX
CORELVER	CORELVER	Correlator software version numb	Observables	CorrInfo_bX
CORPLACE	Correlator	Correlator name.	-	Head
CORR UTC	UTCCorr	UTC time tag of correlation.	Observables	CorrInfo_bX
CORRTYPE	CorrelatorType	Correlator type: MK3/MK4/K4 etc.	-	Head
CROOTFIL	CROOTFIL	Correlator root file name.	Observables	CorrInfo_bX
CT SITE1	FractC	Coordinate time at site 1	Solve	FractC
CTI CFLG	CTIFlag	CTIMG Flow Control Message Der	ObsTheoretical	CalcInfo
CTI MESS	CTIMessage	CTIMG Message Definition	ObsTheoretical	CalcInfo
DBEDITVE	DBEDITVE	Dbedit revision date YYYY MM DD	Observables	CorrInfo_bX
DEL FRQS	DEL FRQS	ALIAS OF REF FREQ.	OBSOLETE	-
DEL OBSV	GroupDelay	Delay observable produced by fringing.	Observables	GroupDelay_bX
DELOBSVM	DELOBSVM	Obs delay at central epoch us.	Observables	CorrInfo_bX
DELRESID	DELRESID	Delay residual (sec).	Observables	CorrInfo_bX
DELSIGMA	GroupDelaySig	Delay Measurement Sigma	Observables	GroupDelay_bX
DELTAEPO	DELTAEPO	Offset from center of scan (sec)	Observables	CorrInfo_bX
DELTFLAG	DELTFLAG	Delay type 1=grp 2=phs.	OBSOLETE	-
DELUFLAG	DelayFlag	Delay unweight flag	ObsEdit	Edit
DISCARD	DISCARD	Percent data discarded by FRNGE.	Observables	CorrInfo_bX
DLERR XS	GroupDelaySig	Delay Measurement Sigma	Observables	GroupDelay_bS
DLOBS XS	GroupDelayFull	Delay Observable with ambiguities resolved and added.	ObsEdit	GroupDelayFull_bS

DLYEPO+1	DLYEPO+1	Phase delay at epoch+1 sec.	Observables	CorrInfo_bX
DLYEPO-1	DLYEPO-1	Phase delay at epoch-1 sec.	Observables	CorrInfo_bX
DLYEPOCH	DLYEPOCH	Phase delay at epoch usec .	OBSOLETE	-
DNP DATA	DNP DATA	Diurnal polar motion scale fact.	OBSOLETE	-
DOMESNR	DomesNumber	Domes number array	Apriori	StationApriori
DPHAS XS	PhaseDelay	Phase delay resolved with ambiguities added.	ObsEdit	PhaseDelayFull_bS
DPHER XS	PhaseSig	Phase delay sigma.	Observables	Phase_bS
DURATION	DURATION	Scan duration (sec).	Observables	CorrInfo_bX
EARTH CE	EarthXYZ	Earth barycentric coordinates.	Scan	Ephemeris_kDE405JPL
EARTHRAD	EARTHRAD	constant	OBSOLETE	-
ECCCOORD	EccentricityVector	Eccentricity taken from eccentricity file.	Apriori	EccentricityApriori
ECCNAMES	EccentricityMonument	Eccentricity monument name	Apriori	EccentricityApriori
ECCTYPES	EccentricityType	Eccentricity type: XY or NE	Apriori	EccentricityApriori
EFF FREQ	DummyName	NOTE: Split into 3 variables.	-	-
EFF.DURA	EffectiveDuration	Effective run duration sec.	Observables	CorrInfo_bX
E-FLAT	E-FLAT	Earth flattening con (unitless).	OBSOLETE	-
EL-CUT	EL-CUT	Solution elevation cutoff data.	OBSOLETE	-
EL-THEO	ElTheo	Elevation array definition	-	AzEl
EMS/MMS	EMS/MMS	The Earth-Moon mass ratio.	OBSOLETE	-
EOPSCALE	EOPSCALE	Time scale of EOP table epochs.	OBSOLETE	-
EPHEPOCH	EPHEPOCH	JPL EPOCH 1950 or 2000 .	OBSOLETE	-
EPNAMES	EPNAMES	Site names for clock-atm epochs	OBSOLETE	-
ERPHS K	PhaseBLWeights	Phase formal error constants	Solve	PhaseBLWeights
ERROR BL	GroupBLWeightName	B.L.names for formal errors	Session	GroupBLWeights
ERROR K	GroupBLWeights	Group delay and rate re-weighting constants.	Session	GroupBLWeights
ERRORATE	ERRORATE	Log err rate by sta sb channel	Observables	CorrInfo_bX
ETD CFLG	EarthTideFlag	Earth Tide flow control mess def	ObsTheoretical	CalcInfo
ETD CONT	EarthTideCal	Earth tide contributions def.	ObsCalTheo	Cal-EarthTide
ETD DATA	EarthTideData	Earth tide module data (la. h l)	ObsTheoretical	CalcInfo
ETD MESS	EarthTideMessage	Earth Tide message definition	ObsTheoretical	CalcInfo

EXPCODE	ExpName	Experiment name.	-	Head
EXPDESC	ExpDescription	Experiment description.	-	Head
EXPNAME	ExpName	Observing program (exp.) name	-	Head
EXPSERNO	ExpSerialNumber	Experiment Serial Number.	-	Head
F12 VOLT	F12VOLT	F1, F2 VOLT .	-	WVR
FALSEDET	ProbFalseDetection	Prob of false det from FRNGE.	Observables	CorrInfo_bX
FCL FLGS	FlybyFlag	Standard flcal configuration	Solve	CalibrationSetup
FCL LIST	FlybyName	Key to the standard flcal config	Solve	CalibrationSetup
FEED.COR	FeedCorrectionCal	Feedhorn corr. in CORFIL scheme	ObsCalTheo	Cal-FeedCorrection
FOURF CF	FourfitControlFile	Control file name for fourfit.	Solve	Misc
FOURF CS	FourFitCmdCString	Command string used for fourfit.	Solve	Misc
FOURFFIL	FOURFFIL	Fourfit output filename.	Observables	CorrInfo_bX
FOURFFXS	FOURFFXS	Fourfit output filename S-band	Observables	CorrInfo_bX
FOURFUTC	FOURFUTC	Fourfit processing time YMDHMS.	Observables	CorrInfo_bX
FOURFVER	FOURFVER	Fourfit version number.	Observables	CorrInfo_bX
FRNGERR	FRNGERR	Fourfit error flag blank=OK.	Observables	CorrInfo_bX
FROTDLAY	FROTDLAY	Delay due to rel. feed rotation.	OBSOLETE	-
FRQGROUP	FRQGROUP	Frequency group code.	Observables	CorrInfo_bX
FUT1 INF	UT1ArrayInfo	Array: (FJD of start, spacing in days, number points, Scaling (should be 1))	Solve	CalcEop
FUT1 PTS	UT1Values	Final Value TAI-UT1 data points.	Solve	CalcEop
FUT1TEXT	UT1Origin	Final Value TAI-UT1 origin text.	Solve	CalcEop
FWOB INF	WobArrayInfo	Array: (FJD of start, spacing in days, number points)	Solve	CalcEop
FWOBTEXT	WobbleOrigin	Final Value wobble origin text.	Solve	CalcEop
FWOBX&YT	WobValues	Final wobble X Y component value	Solve	CalcEop
GAUSS	GAUSS	Gauss's constant (rad).	OBSOLETE	-
GC MBD	GeocMBD	Tot geocenter group delay (sec).	Observables	CorrInfo_bX
GC PHASE	GeocPhase	Tot phase ref to cen of Earth.	Observables	CorrInfo_bX
GC RATE	GeocRate	Tot geocenter delay rate (s/s).	Observables	CorrInfo_bX

GC SBD	GeocSBD	Tot geocenter sbd delay (sec).	Observables	CorrInfo_bX
GCRESPHS	GeocResidPhase	Resid phs corrected to cen of E.	Observables	CorrInfo_bX
GMEARTH	GMEARTH	GM of the Earth (m^3/sec^2).	OBSOLETE	-
GMMOON	GMMOON	GM of the Moon (m^3/sec^2).	OBSOLETE	-
GMSUN	GMSUN	GM of the Sun (m^3/sec^2).	OBSOLETE	-
GPDLAMBG	AmbigSize	Group delay ambiguity spacing	Observables	AmbigSize_bX
GRIONFRQ	FreqGroupIon	Effective Group Delay Ionospheric Frequency	ObsDerived	EffFreq_bX
GROBSDEL	GroupDelay	Observed group delay in sec.	OBSOLETE	-
GRPAMB_S	AmbigSize	Group delay ambiguity spacing	Observables	AmbigSize_bS
HALF PI	HALF PI	Half pi - to 17 digits accuracy.	OBSOLETE	-
IDELAY	IDELAY	Corel instrumental delay (sec).	Observables	CorrInfo_bX
INCOH2	INCOH2	Incoh amp from FRNGE plot segs.	Observables	CorrInfo_bX
INCOHAMP	INCOHAMP	Fr. amp from incoh int of chan.	Observables	CorrInfo_bX
INDEXNUM	INDEXNUM	Corel index numbers by sb freq.	Observables	CorrInfo_bX
INTERVAL	iUTCInterval	First and last UTC time tag in input file.	-	Head
INTRVAL4	iUTCInterval	First and last UTC time tag in input file.	-	Head
ION CODE	IonCode	Ion corr code. -1=no good 0=OK	OBSOLETE	-
ION CORR	IonGroupCal	Ion correction. Add to theo. sec	ObsDerived	Cal-IonGroup_bX
ION_BITS	IonBits	ICORR for full ion tracking.	Solve	IonBits
IONFACTR	IONFACTR	S/X DIFFERENCE ION SCALING FACTO	OBSOLETE	-
IONPCORR	IonPhaseCal	Phase Corr Iono	ObsDerived	Cal-IonPhase_bX
IONRMS	IonGroupCalSigma	Ion correction to sigma. sec	ObsDerived	Cal-IonGroup_bX
IONRMSPH	IonPhaseCalSigma	Phase Corr RMS	ObsDerived	Cal-IonPhase_bX
JUL DATE	JUL DATE	Julian date at midnight.	OBSOLETE	-
LIMSTOPS	LIMSTOPS	Antenna limit stops (deg).	OBSOLETE	-
LO FREQ	LOFreq	LO frequencies per cha/sta MHz.	Observables	CorrInfo_bX
MARI.ATM	MARI.ATM	Marini ATM Model	OBSOLETE	-
MARI.DRY	MARI.DRY	Dry Marini atmosphere	OBSOLETE	-
MARI.WET	MARI.WET	WET COMP OF MARI	OBSOLETE	-
MARISTAT	MARISTAT	Stations with Marini calib	OBSOLETE	-

MCALNAMS	MCALNAMS	Mode calibrations names	OBSOLETE	-
MCALSTAT	MCALSTAT	Mode calibrations status	OBSOLETE	-
MEANCABL	MEANCABL	Mean cable calibration (seconds)	OBSOLETE	-
MOONDATA	MoonXYZ	Lunar geocentric coordinates.	Scan	Ephemeris_kDE405JPL
NB DELAY	NB DELAY	Single band delay (microsec).	OBSOLETE	-
NDRYCONT	NMFDryCal	Nhmf (dry) atm. contribution	-	Cal-NMFDry
NDRYPART	NMFDryPart	Nhmf2 dry partial deriv. def.	-	Part-NMFDry
NGRADPAR	NMFGradPart	Niell dry atm. gradient partials	-	NGRADPAR
NLAGS	NumLagsUsed	Num of lags used for correlation	Solve	Misc
NO.OF AP	NumAp	# of AP by sideband and channel.	Observables	CorrInfo_bX
NUM4 OBS	NumObs	Number of observations (I*4)	-	Head
NUMB OBS	NumObs	Number of observations in file.	-	Head
NUT CFLG	NutationFlag	Nutation flow control mess def.	ObsTheoretical	CalcInfo
NUT MESS	NutationMessage	Nutation message definition	ObsTheoretical	CalcInfo
NUT PART	NutPsiEpsPart	Pre Calc10 Nuation Partial	ObsPart	Part-NutPsiEps
NUT WAHR	NutWahrPsiEps	Wahr nut vals - Dpsi Deps&rates	Scan	Nut_kWahr
NUT2000A	Nut2000PsiEps	IAU200A Nut. - Dpsi Deps Rates	Scan	Nut_k2000PsiEps
NUT2000P	Nut2000PsiEpsPart	IAU2000A Nutation Psi Eps Partls	ObsPart	Part-Nut2KPsiEps
NUT2KXYP	Nut2KXYPart	IAU2000A Nutation X Y Partials	ObsPart	Part-Nut2KXY
NUT2KXYS	Nut2000XYS	CIP Coordinates X Y S and Rates	Scan	Nut_k2000XYS
NWETCONT	NMFWetCal	Whmf (wet) atm. contribution	-	Cal-NMFWet
NWETPART	NMFWetPart	NMF Wet Partial	-	Part-NMFWet
OBCLFLGS	ObsCalFlag	Bit set indicate that calibration is recom-mended.	Solve	CalibrationSetup
OBCLLIST	ObsCalName	Available obs dependent calibrations (pole-tide, earthdide, …)	Solve	CalibrationSetup
O-C RES	O-C RES	DELAY RATE RESIDUALS(s, s per s)	OBSOLETE	-
OCAMP EW	OceanAmpEW	Ocean amp. E-W component(meters)	ObsTheoretical	CalcInfo
OCAMP NS	OceanAmpNS	Ocean amp. N-S component(meters)	ObsTheoretical	CalcInfo
OCAMP RD	OceanAmpUp	Ocean amp. radial comp.-meters.	ObsTheoretical	CalcInfo

OCCUPNUM	OCCUPNUM	Site Occupation Number.	Observables	CorrInfo_bX
OCE CFLG	OceanFlag	Ocean load flow control mess def	ObsTheoretical	CalcInfo
OCE CFLG	OceanFlag	Ocean load flow control mess def	ObsTheoretical	CalcInfo
OCE CFLG	OceanFlag	Ocean load flow control mess def	ObsTheoretical	CalcInfo
OCE CONT	OceanLoadCal	Obs dependent ocean loading	ObsCalTheo	Cal-OceanLoad
OCE DELD	OceanLoadDis	Ocean load site depndnt displace	-	Dis_kOceanLoad
OCE HORZ	OceanLoadHorizCal	Site-dep ocean cont - horizontal	-	Cal-OceanLoad
OCE MESS	OceanMessage	Ocean loading message definition	ObsTheoretical	CalcInfo
OCE STAT	OceanStations	Ocean loading station status.	ObsTheoretical	CalcInfo
OCE STAT	OceanStations	Ocean loading station status.	ObsTheoretical	CalcInfo
OCE STAT	OceanStations	Ocean loading station status.	ObsTheoretical	CalcInfo
OCE VERT	OceanLoadVertCal	Site-dep ocean cont - vertical	-	Cal-OceanLoad
OCPHS EW	OceanPhaseEW	Ocean phase E-W component (rad)	ObsTheoretical	CalcInfo
OCPHS NS	OceanPhaseNS	Ocean phase N-S component (rad)	ObsTheoretical	CalcInfo
OCPHS RD	OceanPhaseUp	Ocean phase radial comp.-radians	ObsTheoretical	CalcInfo
ORIGFILE	ORIGFILE	Original COREL file name.	Observables	CorrInfo_bX
PAN MESS	FeedhornMessage	Feedhorn rot. angle mod. ident.	ObsTheoretical	CalcInfo
PARANGLE	FeedRotation	Feedhorn rotation angle.	-	FeedRotation
PEP MESS	PepMessage	PEP Utility Message Definition	ObsTheoretical	CalcInfo
PHASECAL	PHASECAL	PC rate by sta (us per s).	Observables	CorrInfo_bX
PHCALOFF	PhaseCalOffset	Phase cal offset (-18000/18000).	Observables	CorrInfo_bX
PHCALSTS	PHCALSTS	Phase cal status: 0/1/2 .	OBSOLETE	-
PHIONFRQ	FreqPhaseIon	Effective Phase Delay Ionospheric Frequency	ObsDerived	EffFreq_bX
PHSUFLAG	PhaseFlag	Phase unweight flag	ObsEdit	Edit
PI	PI	Pi - to 17 digits accuracy.	OBSOLETE	-
PI NAME	PrincipalInvestigator	Agency/contact_person/PI name.	-	Head
PLX CFLG	ParallaxFlag	Parallax flow control mess def	ObsTheoretical	CalcInfo
PLX MESS	ParallaxMessage	Parallax message definition	ObsTheoretical	CalcInfo
PLX PART	ParallaxPart	Parallax partial deriv. def.	ObsPart	Part-Parallax
PLX1PSEC	ParallaxCal	Parallax partial/contr 1 parsec	ObsCalTheo	Cal-Parallax

POLAR XY	PolarMotion	Polar motion X & Y for obs (rad)	Scan	EOPApriori
POLARIZ	Polarization	Polarization per sta/chan R/L.	Observables	CorrInfo_bX
PRE CFLG	PrecessionFlag	Precession flow contril mess def	ObsTheoretical	CalcInfo
PRE DATA	PrecessionData	Precession constant (asec/cent).	ObsTheoretical	CalcInfo
PRE MESS	PrecessionMessage	Precession message definition	ObsTheoretical	CalcInfo
PRE PART	PrecessionPart	Precession partial deriv. def.	ObsPart	Part-Precession
PROC UTC	UTCProc	YDDD of COREL by sta channel.	Observables	CorrInfo_bX
PRT FLAG	PRT FLAG	Atmosphere partial to be applied	OBSOLETE	-
PTD CFLG	PoleTideFlag	Pole tide flow control mess def	ObsTheoretical	CalcInfo
PTD CONT	PoleTideCal	Pole tide contributions def.	ObsCalTheo	Cal-PoleTide
PTD MESS	PoleTideMessage	Pole tide message definition	ObsTheoretical	CalcInfo
PTDXYPAR	WobPart	Pole Tide Partials w.r.t. X & Y	ObsPart	Part-Poletide
PTOLDCON	PoleTideOldRestoreCal	Old Pole Tide Restorer Contrib.	ObsCalTheo	Cal-PoleTideOldRestore
QBFACTOR	QBFACTOR	Measure of uniformity of data.	Observables	CorrInfo_bX
QCODE XS	QualityCode	FRNGE quality index 0 --> 9	Observables	QualityCode_bS
QUALCODE	QualityCode	FRNGE quality index 0 --> 9	Observables	QualityCode_bX
RAD ASEC	RAD ASEC	Radians per arc second .	OBSOLETE	-
RAD DEG	RAD DEG	Radians per degree .	OBSOLETE	-
RAD TSEC	RAD TSEC	Radians per time second .	OBSOLETE	-
RAD/ASEC	RAD/ASEC	Radians per arc second .	OBSOLETE	-
RAD/DEG	RAD/DEG	Number of radians per degree.	OBSOLETE	-
RAD/TSEC	RAD/TSEC	Radians per time second .	OBSOLETE	-
RAT FRQS	RAT FRQS	ALIAS OF REF FREQ.	OBSOLETE	-
RAT OBSV	GroupRate	Rate Observable	Observables	GroupRate_bX
RATOBSVM	RATOBSVM	Obs rate at central epoch .	Observables	CorrInfo_bX
RATRESID	RATRESID	Rate resid (sec per sec).	Observables	CorrInfo_bX
RATSIGMA	GroupRateSig	Rate Measurement Sigma	Observables	GroupRate_bX
RATUFLAG	RateFlag	Delay rate unweight flag.	ObsEdit	Edit
RECMODE	RecordingMode	Recoding mode.	-	Head

RECSETUP	RECSETUP	Samp rate(kHz) Frames/PP PP/AP.	Observables	CorrInfo_bX
RECTRACK	RECTRACK	Trk table by sta sideb channel.	Observables	CorrInfo_bX
REF FR_S	RefFreq	Frequency to which phase is referenced	Observables	RefFreq_bS
REF FREQ	RefFreq	Frequency to which phase is referenced	Observables	RefFreq_bX
REFCLKER	REFCLKER	Ref sta clock epoch microsec.	Observables	CorrInfo_bX
REL CFLG	RelativityFlag	Relativisitc bending use status	ObsTheoretical	CalcInfo
REL DATA	RelativityMessage	Relativity mod data (gamma).	ObsTheoretical	CalcInfo
REL.HUM.	RelHum	Rel.Hum. at local WX st (50%=.5)	-	Met
RFREQ	ChannelFreq	RF freq by channel (MHz).	Observables	CorrInfo_bX
ROTEPOCH	ROTEPOCH	UT1 pole interp.reference epochs	OBSOLETE	-
RTERR XS	GroupRateSig	Rate Measurement Sigma	Observables	GroupRate_bS
RTOBS XS	GroupRate	Rate Observable	Observables	GroupRate_bS
RUN CODE	RunCode	Run code e.g. "329-1300".	Observables	CorrInfo_bX
S2EFFREQ	S2EFFREQ	Effective group freq for ion.	Observables	CorrInfo_bX
S2FLAG	S2FLAG	S2 or MKIII Flag	OBSOLETE	-
S2PHEFRQ	S2PHEFRQ	Effective phase frequency	Observables	CorrInfo_bX
S2REFREQ	S2REFREQ	Effective frequency for rate	Observables	CorrInfo_bX
SAMPLRAT	SampleRate	Sample rate (Hz).	Observables	CorrInfo_bX
SB DEL_S	SBDelay	Single band delay	Observables	SBDelay_bS
SB DELAY	SBDelay	Single band delay	Observables	SBDelay_bX
SB SIG_S	SBDelaySig	Single band delay error	Observables	SBDelay_bS
SB SIGMA	SBDelaySig	Single band delay error	Observables	SBDelay_bX
SBRESID	SBRESID	Single band delay residual.	Observables	CorrInfo_bX
SCAN UTC	UTCScan	Nominal scan time YMDHMS.	Observables	CorrInfo_bX
SCANNAME	ScanName	Scanname in database	Scan	ScanName
SEC TAG	Second	Seconds part of UTC TAG.	Observables	TimeUTC
SIT ELEV	SIT ELEV	Height above ellipsoid (meters)	OBSOLETE	-
SIT MESS	SiteMessage	Site Module Message Definition	ObsTheoretical	CalcInfo
SIT PART	XYZPart	Site partials: dtau/dr_1=-dtau/dr_2	ObsPart	Part-XYZ
SIT_KILL	SIT_KILL	site_killer version date	OBSOLETE	-

SITCODES	SITCODES	Two-letters site codes .	OBSOLETE	-
SITERECS	StationXYZ	Site cartesian coords (m).	Apriori	StationApriori
SITEZENS	SiteZenithDelay	Site zenith path delays (nsec).	ObsTheoretical	CalcInfo
SITHOCAM	OceanHorizontalAmp	Horz ocean loading ampltudes (m)	ObsTheoretical	CalcInfo
SITHOCPH	OceanHorizontalPhase	Horz ocean loading phases (rad).	ObsTheoretical	CalcInfo
SITNAMES	StationNameApriori	Site names in alphabetical order	Apriori	StationApriori
SITOCAMP	OceanUpAmp	Vert ocean loading ampltudes (m)	ObsTheoretical	CalcInfo
SITOCPHS	OceanUpPhase	Vert ocean loading phases (rad).	ObsTheoretical	CalcInfo
SLEWRATS	SLEWRATS	Antenna slew rates (rad/sec).	OBSOLETE	-
SNR_S	SNR	Signal to noise ratio for S-band	Observables	SNR_bS
SNRATIO	SNR	Signal to noise ratio.	Observables	SNR_bX
SOL_DATA	SOL_DATA	Stnd sol data. IDATYP from SOCOM	OBSOLETE	-
SOLVE_VR	SOLVE_VR	SOLVE system version number	OBSOLETE	-
SOURSTAT	SourceSelectionFlag	Source selection status bit-mapped array.	Solve	SelectionStatus
SRCHPAR	SRCHPAR	FRNGE/Fourfit search parameters.	Observables	CorrInfo_bX
STAR ID	Source	Radio source name.	Observables	Source
STAR REF	SourceReference	Source of coordinate values.	Apriori	SourceApriori
STAR2000	Source2000RaDec	J2000 Source RA and Dec	Apriori	SourceApriori
STARELEV	STARELEV	Elev angles calc by COREL.	Observables	CorrInfo_bX
STARTOFF	StartOffset	Offset nominal start time (sec).	Observables	CorrInfo_bX
STARTSEC	STARTSEC	Start time in sec past hour.	Observables	CorrInfo_bX
STAT_ACM	ClockAprioriSite	Stations with a priori clock mod	Apriori	ClockApriori
StationNameCrossRef	StationNameCrossRef	Site names in alphabetical order	CrossReference	StationCrossRef
STOP OFF	StopOffset	Offset nominal stop time (sec).	Observables	CorrInfo_bX
STOP SEC	StopSec	Stop time in sec past hour.	Observables	CorrInfo_bX
STR CFLG	StarParallaxFlag	Parallax flow control mess def	ObsTheoretical	CalcInfo
STR MESS	StarMessage	Star module message definition	ObsTheoretical	CalcInfo
STR PART	RaDecPart	Star partial derivatives def.	ObsPart	Part-RaDec
STR TIME	TimeSinceStart	Interval since start time (sec).	Observables	CorrInfo_bX
STRNAMES	SourceNameApriori	Source names in RA order.	Apriori	SourceApriori

SUN CONT	BendSunCal	Consensus bending contrib. (sec)	ObsCalTheo	Cal-BendSun
SUN DATA	SunXYZ	Solar geocentric coordinates.	Scan	Ephemeris_kDE405JPL
SUN2CONT	BendSunHigherCal	High order bending contrib.(sec)	ObsCalTheo	Cal-BendSunHigher
SUPMET	SUPMET	Suppression method	OBSOLETE	-
TAI- UTC	LeapSecond	Leap second	Solve	Misc
TAPEID	TAPEID	Raw data tape ID for ref and rem	Observables	CorrInfo_bX
TAPQCODE	TapeCode	Tape quality code.	Observables	CorrInfo_bX
TECTMODL	TECTMODL	Default tectonic plate modelname	OBSOLETE	-
TECTPLNM	TECTPLNM	4-char tectonic plate names.	OBSOLETE	-
TEMP C	TempC	Temp in C at local WX station	-	Met
THE MESS	TheoryMessage	Theory module identification	ObsTheoretical	CalcInfo
TIDALUT1	TidalUT1Flag	Flag for tidal terms in UT1 sers	ObsTheoretical	CalcInfo
TILTRMVR	TiltRemoverCal	Axis Tilt Contribution Remover	ObsCalTheo	Cal-TiltRemover
TOTPCENT	TOTPCENT	Tot phase at central epoch.	Observables	CorrInfo_bX
TOTPHA_S	Phase	Total phase	Observables	Phase_bS
TOTPHASE	Phase	Total phase	Observables	Phase_bX
TPHA ERR	TotalFringeErr	Total fringe phase error (deg)	Observables	CorrInfo_bX
TSEC/AU	TSEC/AU	Number of seconds per A.U.	OBSOLETE	-
TSEC/DAY	TSEC/DAY	Second per day .	OBSOLETE	-
TSYS TMP	Tsys	TSYS TMP (K) 14CH AND IF1 IF2	Observables	Tsys
TWOPI	TWOPI	TWO PI .	OBSOLETE	-
UACSUP	UserSup	User action for suppression	ObsEdit	Edit
U-GRV-CN	U-GRV-CN	Universal grav con m^3/kg*s^3.	OBSOLETE	-
UNCALDLY	UNCALDLY	Bad Phase Cal Delay. Removed from Obs.	-	-
UNCALFLG	UNCALFLG	Delay+rate uncal if flag=1.	OBSOLETE	-
UNPHAFLG	UNPHAFLG	UnPhaseCal flag	Solve	UnPhaseCal
UNPHASCL	UnPhaseCal	UnPhaseCal effect - group&rate	ObsCalTheo	Cal-UnphaseCal
URVR	URVR	Rate derivatives mHz per asec.	Observables	CorrInfo_bX
USER_REC	USER_REC	User observation recovery status	OBSOLETE	-
USER_SUP	USER_SUP	User observation suppression sts	OBSOLETE	-

UT1 CFLG	UT1FLag	UT1 control flag message def.	ObsTheoretical	CalcInfo
UT1 MESS	CalcUt1Module	UT1 Module message definition	Solve	CalcEop
UT1 PART	UT1Part	UT1 partial derivatives def.	ObsPart	Part-EOP
UT1 -TAI	UT1	UT1 time of day for this obsvr.	Scan	EOPApriori
UT1EPOCH	UT1EPOCH	TAI - UT1 epoch value definition	ObsTheoretical	CalcInfo
UT1INTRP	UT1IntrpMode	Message for UT1 interp. scheme	Solve	CalcEop
UT1ORTHO	UT1OrthoCal	ORTHO_EOP Tidal UT1 contribution	ObsCalTheo	Cal-HFEOP-IERS2003
UTC ERR	UTCErr	A priori UTC error site 1 (sec)	Observables	CorrInfo_bX
UTC TAG	YMDHM	Epoch UTC YMDHM.	Observables	TimeUTC
UTC TAG4	UTC TAG4	Epoch UTC YMDHMS (4 digit year).	OBSOLETE	-
UTCM TAG	UTCMidObs	UTC at central epoch YMDHMS.	Observables	CorrInfo_bX
UVF/ASEC	UVFperAsec	U V in FR per arcsec from CALC	Solve	UVFperAsec
VFDWELL	VFDWELL	Dwell time in each channel (sec)	Observables	CorrInfo_bX
VFRQ#APS	NumAccum	No. of accum. periods in Channel	Observables	CorrInfo_bX
VFRQAM	VFRQAM	Normalized channel amplitude	Observables	CorrInfo_bX
VFRQPCAM	VFRQPCAM	Phase cal tone Amplitudes	Observables	CorrInfo_bX
VFRQPCFR	VFRQPCFR	Phase cal tone Frequencies	Observables	CorrInfo_bX
VFRQPCPH	VFRQPCPH	PHASE cal tone Phases	Observables	CorrInfo_bX
VFRQPH	VFRQPH	Channel Phase (degrees)	Observables	CorrInfo_bX
VIRTFREQ	VIRTFREQ	Sky Frequencies	Observables	CorrInfo_bX
VLB1FILE	VLB1FILE	Correlator file name.	Observables	CorrInfo_bX
VLB1XTNT	VLB1XTNT	corr. ext by sideb channel.	Observables	CorrInfo_bX
VLB2 UTC	UTCVLB2	UTC of FRNGE proc YMDHMS.	Observables	CorrInfo_bX
VLB2PRG	VLB2PRG	FRNGE(YYMMDD) Fourfit(x.x) ver.	Observables	CorrInfo_bX
VLB2XTNT	VLB2XTNT	FRNGE extent number.	Observables	CorrInfo_bX
VLIGHT	VLIGHT	The speed of light (m/sec).	OBSOLETE	-
WAHRCONT	NutWahrCal	2000A Nut to Wahr Nut Contributn	ObsCalTheo	Cal-NutWahr
WBEPOCH	WBEPOCH	LIST OF SITE NMES FOR EPOCHS	OBSOLETE	-
WOB CFLG	WobbleFlag	Wobble flow control mess def.	ObsTheoretical	CalcInfo
WOB MESS	CalcWobModule	Wobble message definition.	Solve	CalcEop

WOB PART	WobblePart	Wobble partial derivatives def.	ObsPart	Part-EOP
WOBEPOCH	WOBEPOCH	Interpolated wobble array def	ObsTheoretical	CalcInfo
WOBINTRP	WobIntrpMode	Interp. scheme for polar motion.	Solve	CalcEop
WOBNUTAT	WobNutCal	Short period nutation wobble con	ObsCalTheo	Cal-WobNut
WOBORTHO	WobOrthoCal	ORTHO_EOP tidal wobble contribtn	ObsCalTheo	Cal-HFEOP-IERS2003
WOBXCONT	XwobbleCal	X Wobble contribution definition	ObsCalTheo	Cal-Wobble
WOBYCONT	YwobbleCal	Y Wobble contribution definition	ObsCalTheo	Cal-Wobble
WVR CODE	WVRQcode	WVR DELAY DATA QUALITY CODE	-	WVR
WVR DELY	WVRDelay	WVR LINE OF SIGHT & ZENITH(NSEC)	-	WVR
WVR STAT	WVR STAT	Stations with WVR calib .	OBSOLETE	-
WVR TEMP	WVRTempRef	REFERENCE1,2 WVR TEMP (K).	-	WVR
WVR.DELY	WVRDelay	WVR PATH DELAYWVR PATH DELAY AND	-	WVR
WVRDELAY	WVRDelay	WVR line of sight & ZENITH(sec)	-	WVR
ZDELAY	ZDELAY	Corel zenith atmos. delay (sec).	Observables	CorrInfo_bX