

Report on IVS-WG4

John Gipson

Abstract In 2007 the IVS Directing Board established IVS Working Group 4 on VLBI Data Structures. This note discusses the history of WG4, presents an overview of the proposed structure, and presents a timeline for next steps.

Keywords IVS Working Group IV, netCDF *Calc/Solve*, *Mark3* databases, VLBI processing

1 Introduction

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 website Gipson (2007): “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.”

John Gipson
NVI, Inc./NASA Goddard Spaceflight Center, Greenbelt, MD,
20770, USA

Chair	John Gipson
Analysis Center Director	Axel Nothnagel
Correlator Representatives	Roger Cappalo Colin Lonsdale
GSFC/Calc/Solve	David Gordon Leonid Petrov
JPL/Modest	Chris Jacobs Ojars Sovers
Occam	Oleg Titov Volker Tesmer
TU Vienna	Johannes Boehm
IAA	Sergey Kuurdobov
Steelbreeze	Sergei Bolotin
Observatoire de Paris/PIVEX	Anne-Marie Gontier
NICT	Thomas Hobiger Hiroshi Takiguchi

Table 1 Original Membership in Working Group 4

Changes to the VLBI data format affect everyone in the VLBI community. Hence it is important that the Working Group have representatives from a broad cross-section of the IVS community. Table 1 lists the original members of IVS WG4 together with their original affiliations.¹ The initial membership was arrived at in consultation with the IVS Directing Board. On the one hand, we wanted to ensure that all points of view were represented. On the other hand, we 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.

¹ Membership was subsequently reduced for various reasons: Colin Lonsdale resigned because of increased management responsibilities; Leonid Petrov left the Goddard VLBI group; And, most sadly the premature death of Anne-Marie Gontier.

Table 2 Key Goals of New Format

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	Commonly used data should be able to be retrieved quickly
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. Most users are interested only 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

2 History of Working Group 4

WG4 held its first meeting at the 2008 IVS General Meeting in St. Petersburg, Russia. This meeting was open to the IVS community. Roughly 25 scientists attended: 10 WG4 members, and 15 others. This meeting was held after a long day of proceedings. The number of participants and the ensuing lively discussion 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 lots more information Gipson (2008).

During the next year the WG4 communicated via email and telecon and discussed how to meet the goals that emerged from the St. Petersburg meeting. A consensus began to emerge.

The next face-to-face meeting of WG4 was held at the 2009 European VLBI 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. Concurrently I 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 particular subset included all data in NGS cards and a little more. The subset was chosen because many VLBI analysis packages including Occam, Steelbreeze, and VIEVS can use NGS cards as input. In August 2009 we made available, via anonymous ftp, three VLBI sessions in the new format: an Intensive, an R1 and an RDV.

3 Overview of New Organization

In a paper as brief as this it is impossible to completely describe the new organization and format. Instead I briefly describe three of the key components: 1) Modularization; 3) Organize data through wrappers; 2) Storing data in netCDF files;

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; station positions; etc. In many, though not all, cases, each 'type' corresponds to a Mark3 database L-code. 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 remaining are other advantages not

originally specified, but are consequences of this design decision.

1. **Separable.** Users can retrieve only that part of the data they are interested.
2. **Extensible.** It is easy to add new data-types by specifying the data and file format for the new data.
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. **Flexible.** Since different data is kept in different files, it is easy to add new data types.
5. **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 changed²

Data is also organized by 'scope', that is how broadly applicable it is: Does it hold for the entire session, for a particular scan, for a particular station and station, or for a particular observation? Mark3 databases are observation oriented: all data is stored once for each observation. This results in tremendous redundancy for some data. For example, consider an N -station scan, with $(N - 1) \times N/2$ observations, with each station participating in $N - 1$ observations. Station dependent data, such as met or pointing data, will be the same for all observations involving a given station. Storing this data once per observation instead of once per scan results in an $(N - 1)$ fold redundancy.

3.2 Organizing Data by Wrappers

The main disadvantage of breaking up the VLBI data into many smaller files is that you need some way of organizing the files. This is where the concept of a wrapper comes in. A wrapper is an ASCII file that contains pointers to VLBI files associated with a session. VLBI analysis software parses this file and reads in the appropriate data. As new data types are added, or as data is updated, new versions of the wrapper are generated. The wrapper concept is illustrated schematically in 2. The wrapper can serve several different purposes:

1. The wrapper can be used by analysis programs to specify what data to use.

² This would be done by making a new version of the relevant file, keeping the old one intact.

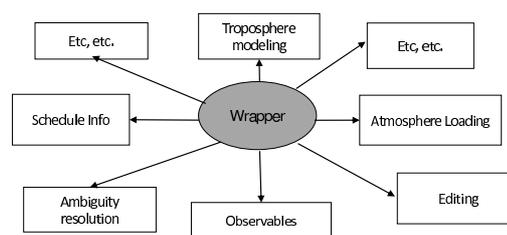


Fig. 1 Wrappers organize the data.

2. Wrappers allows analysts to experiment with 'what if' scenario. For example, to use another analysts editing criteria all you need to do is modify the wrapper to point to the alternative editing file.
3. Because of the general structure of the wrapper, different analysis packages can use different wrappers that point to different subsets of the VLBI data.
4. The wrapper is a convenient means of signaling to the IVS data center what information is required. In this scenario, a user writes a wrapper with pointers to the relevant files and sends it to the IVS data center. The data center packages the data in a tar file and makes it available.

3.3 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 since 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. Quoting from the Unidata web-site:³

NetCDF (network Common Data Form) is a set of interfaces for array-oriented data access and a freely-distributed collection of data access libraries for C, Fortran, C++, Java, and other languages. The netCDF libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data.

NetCDF data is:

³ www.unidata.ucar.edu/software/netCDF/docs/faq.html#whatitis

- **Self-Describing.** A netCDF file includes information about the data it contains.
- **Portable.** A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- **Scalable.** A small subset of a large dataset may be accessed efficiently.
- **Appendable.** Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- **Sharable.** One writer and multiple readers may simultaneously access the same netCDF file.

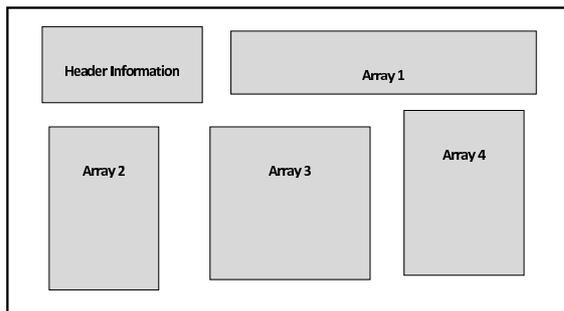


Fig. 2 A netCDF file is a container for arrays.

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 types (in a programming language sense) of data – strings, integer, real, double, etc. Since most VLBI data is some kind of array using netCDF is a natural choice. These files can contain history entries which aid in provenance. Storing data in netCDF format has the following advantage:

1. **Platform/OS/Language Support.** NetCDF has interface libraries to all commonly used computer languages running on a variety of platforms and operating system.
2. **Speed.** NetCDF is designed for fast data access.
3. **Compactness.** 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. **Large User Community.** There are many freely available programs to work with netCDF files.

Because of the open architecture of this system, I propose calling the new format “VLBI OpenDB Format”, or OpenDB for short.

4 Data Transition and Calc/Solve Issues

The starting point for most IVS-analysis packages is a “Version 4” Mark3-database⁴. A Version 4 database has all the ambiguities resolved and the data is edited to flag bad data. Version 4 databases are produced by *Calc/Solve*⁵. Hence any discussing of transitioning from Mark3 databases to OpenDB format must deal with the *Calc/Solve* transition as well.

It is useful to have an understanding of the key stages in the transformation of VLBI correlator output to a Version 4 database ready for distribution. The following describes the processing at Goddard. The details may differ at other institutions, but the fundamentals remain the same. Anytime data is added to a Mark3 database a new version is produced.

1. For each band *Dredit* makes a Version 1 database from the correlator output. Typically this is X- and S-band, although a few sessions use other bands.
2. *Dbscal* inserts cable-cal and met, making a Version 2 database. Cable-cal and met data are used by most analysis packages.
3. *Calc* computes partials and a priori, and creates a Version 3 database. In contrast to cable-cal and met data, although much of this information is required by *Solve*, it is not used by other analysis package.
4. *Interactive-Solve* is used to resolve ambiguities, edit the data, apply ionosphere corrections, and merge the X- and S-band database together. The Version 4 database is ready for distribution.
5. Many analysis packages use so-called “NGS cards”. This is ASCII representation of a subset of the data in a Mark3 database. *Db2ngs* extracts and converts the data from the database.

By design, the Mark3 database contains almost all of the data required to analyze VLBI data.⁶ However reading a Mark3 database is very slow. Because of this the Goddard VLBI group developed “superfiles” which contain a subset of the Mark3 database in binary form. Superfiles can be used in *Interactive-Solve*, but are typically used in *Global-Solve* which “stacks” individual sessions together to obtain, for example, estimates of station position and velocity, or source positions.

⁴ Perhaps in NGS card format.

⁵ Here “*Calc/Solve*” refers to the entire suite of software developed to process and analyze Mark3 databases.

⁶ A few notable exceptions include EOP, atmospheric loading, VMF data, and information about breaks in station position.

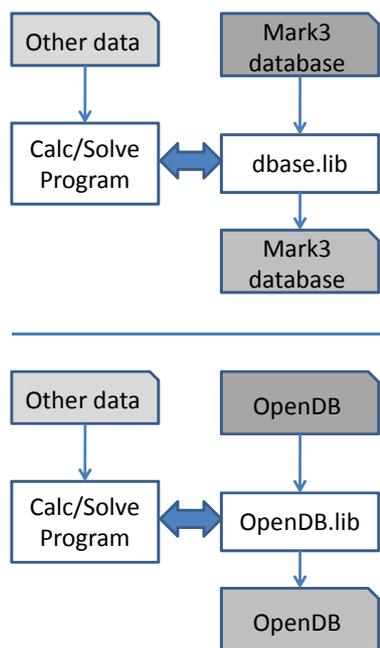


Fig. 3 Replacing dbase.lib with OpenDB.lib allows us to almost transparently produce OpenDB files.

Since *Calc/Solve* is used to produce Version 4 databases, the IVS cannot completely transition to the OpenDB format until *Calc/Solve* is modified to handle it. A serious obstacle to modifying *Calc/Solve* is that it is operational software. We must make sure that it continues to work, and that at all stages we maintain continuity with previous versions. The conversion of *Calc/Solve* to use the new format is taking place in several steps.

1. To maintain compatibility with the current software, we are developing a replacement for the database library which will read and write OpenDB format. In terms of *Calc/Solve* the function calls will be identical. This minimizes changes to existing programs. This is illustrated schematically in Figure 3.
2. In the Summer of 2011 we will complete *db2OpenDB* which converts a Mark3 database into OpenDB format. Originally this just converted the data contained in NGS cards. Currently it converts about 90% of the data⁷ in Mark3 databases.
3. I am modifying *Global-Solve* to use OpenDB format instead of superfiles. This process, begun in

⁷ This excludes some items obsolete or unused items such as the numerical value of π , or the speed of light.

Fall 2010, should be also be complete in Summer 2011. Preliminary results indicate that there will be little, if any timing penalty. There may be even be a slight performance boost because of reduced I/O.

5 Next Steps

In the previous section I discussed the status of the *Calc/Solve* analysis software. Below I summarize the status of some other analysis packages.

- In the Fall of 2009 and in the Spring of 2010 the VLBI group at the Technical University of Vienna developed the interface to *VieVs*.
- Oleg Titov has begun re-writing Occam to use the new format.
- Thomas Hobiger has indicated that C5++ will be modified to use the new format.

In terms of transitioning to the new format:

1. We will make one year of VLBI data available in OpenDB format in July 2011. This will give software-developers something to work with.
2. Interfacing to the new format will give real world experience and may lead to fine-tuning of the specifications. The final specifications will be ready in Fall 2011.
3. In 2012 we will make available all VLBI data in OpenDB format.
4. We will present the final report of IVS WG4 at the 2012 General Meeting.

After the 2012 IVS General Meeting Working 4 will dissolve.

References

1. J. Gipson. <http://ivsc.gsfc.nasa.gov/about/wg4/index.html>, 2007.
2. J. Gipson. IVS Working Group 4 on VLBI Data Structures. *The 5th IVS General Meeting Proceedings*, 2008.
3. J. Gipson. IVS Working Group 4: VLBI Data Structures. *The 6th IVS General Meeting Proceedings*, 2010.