

GBT Data Archive Proposal: Storage, Meta-data, and Access

Dana S. Balser, Brian R. Kent, & Gareth Hunt

21 February 2012

ABSTRACT

The GBT data archive consists of multiple copies of raw GBT data stored on disk and tape in Green Bank and Charlottesville. Meta-data are contained within the archive but are incomplete and in some cases incorrect. Currently there is no archive search and retrieval user interface for the GBT. Here we propose to incorporate the GBT data into the NRAO archive access tool (AAT). We discuss modifications to the data storage structure, the meta-data, and mechanisms to support GBT archive search and retrieval from the AAT.

1. Introduction

Currently the GBT data archive is really a data backup where data are stored on disk and tape in Green Bank and Charlottesville. A real data archive has indexes and search capabilities so that files can be easily located and retrieved. This requires meta-data that accurately describe the data and can be effectively used by search tools. A good user interface is essential to allow users to easily search and retrieve data. An archive should contain all of the information necessary for data analysis software, including pipelines. This includes ancillary data (e.g., weather information) and logs (e.g., operator logs) that are either necessary for the data processing or useful to better understand data in the archive.

Here we propose changes to the data storage structure and meta-data so that the NRAO archive access tool (AAT) can be used to search and retrieve GBT data. This is not a software requirements document but a proposal on how to create a GBT data archive by leveraging existing tools. The intention is that software requirements can quickly be written using this document.

2. Data Storage

2.1. Current GBT Data Storage Structure

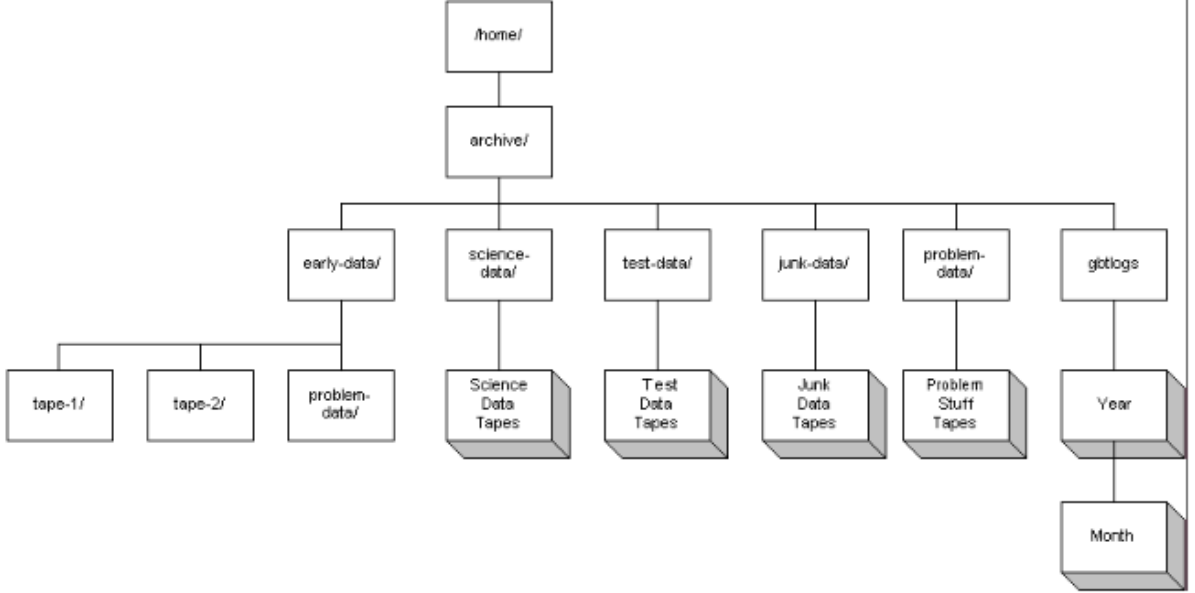


Fig. 1.— Current GBT data storage directory structure. See <http://www.gb.nrao.edu/newgbarchive/>.

Figure 1 illustrates the current directory structure in Green Bank (GB) for the GBT data storage. About every three months data from `/home/gbtdata` are copied to `/home/archive`. These data consist of device specific FITS files for each GBT project. For archive access we are interested in the early science data listed in the sub-directory “early-data” and the rest of the science data listed in the sub-directory “science-data”. Test data, located in the directory test-data, might be of scientific use but there are many subtleties with these data that may be difficult to detect. For example, during spectral baseline and pointing tests the sub-reflector was sometimes adjusted using scripts and thus the telescope was purposely not in focus. This is not obvious to detect and requires a comparison of the encoder values with the focus tracking model. Therefore, we recommend that meta-data for archive access only be generated for science data. Historically these data were backed up on tapes and thus the sub-directory names are tape-0001, tape-0002, etc. Once a new sub-directory was created the previous sub-directory would become static. As of 2011 there are 36 sub-directories in the `/home/archive/science-data` area.

Ancillary data are stored in `/home/gbtlogs`. These data consist of device specific FITS

files that are logged from various instrumentation (e.g., weather station) and are stored as a function of time. Every several months these data are copied to /home/archive/gbtlogs and organized by year and month.

A subset of data in GB have been mirrored to charlottesville (CV) over the intranet for about the past 5 years. To our knowledge the details of this process and the data stored have not been well documented. Currently the early-data, science-data, test-data, and problem-data are mirrored each night along with the current GBT data in /home/gbtdata. A rsync command is executed in GB each night to mirror the data to CV.

Not all GBT science data are archived. Because of the large data sets pulsar data are typically not archived by the NRAO. The exceptions are the pulsar surveys and the NANOGrav project. Other external user back-ends such as the Radar back-end are also not archived.

2.2. Proposed GBT Data Storage Structure

We propose to restructure the data storage in GB and CV to more easily access these data. Figure 2 summarizes the top level directory structure in GB and CV. The arrows indicate data flow. On time scales of about 3 months data will be copied from /home/gbtdata and /home/gbtlogs to /home/archive. Each night data will be copied from seven directories in GB to CV as indicated by the solid line with arrows. This will be performed with an rsync command in GB without using the “–delete” option. Therefore, any data problem in GB will not be propagated to CV. Each night logs of the rsync result should be emailed to system administrators in GB and CV and the authors. In detail:

1. GB (/home/gbtdata) to CV (/export/cvlustre/vulcan/GBT/new-data). Data here are from about the past three months and include both science and test data. We propose that the directory be flat and only include sub-directories for each distinct observing session (e.g. AGBT11B_043_05).
2. GB (/home/archive/early-data) to CV (/export/cvlustre/vulcan/GBT/early-data). Data here are mostly from 2001 and the content should not change in time. We propose that the directory be flat and only include sub-directories for each distinct observing session.
3. GB (/home/archive/science-data) to CV (/export/cvlustre/vulcan/GBT/science-data). All science data in /home/gbtdata are copied here. Science data are typically defined

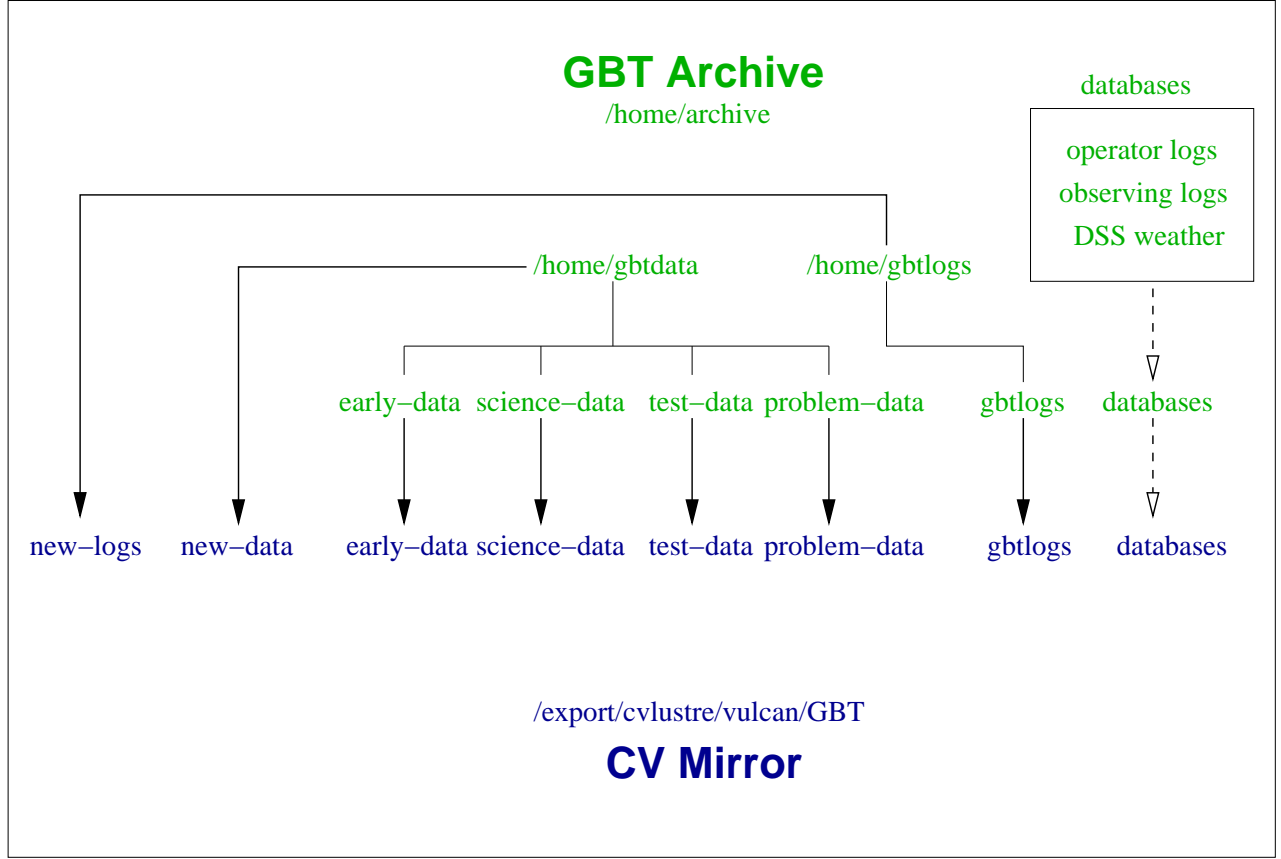


Fig. 2.— Proposed top level GBT data storage directory structure in GB (green) and CV (blue). The solid lines with arrows indicate data copy via rsync and the dotted lines with open arrows indicate database mirrors.

by a session name beginning with an “A”. While this is mostly the case there are some projects, especially for earlier data, that do not conform to this convention.

We propose that a session name of the form AGBTyys_nnn_ss (e.g., AGBT11A_043_23) be placed in a sub-directory “yys” (e.g., 11A). Here “yy” is the year (e.g., 2011) and “s” is the semester (e.g., “A”). All other data should be placed in a sub-directory called “Other”. This would include VLBI sessions which use the legacy proposal code (e.g., ABL175_02) or proposals that have been submitted outside the NRAO (e.g., Fermi, Chandra, etc.).

4. GB (/home/archive/test-data) to CV (/export/cvlustre/vulcan/GBT/test-data). All test data in /home/gbtdata are copied here. Test data may include pointing (e.g., TPTCSPNT_70112), holography (e.g., TPTCSHOL_110330), etc. In some cases observations were performed to test the observing mode of a specific science project (e.g.,

TGBT04A_029.01). These are still located under test-data. The session name should begin with a “T”. We propose that the directory be flat and only include sub-directories for each distinct session.

5. GB (/home/archive/problem-data) to CV (/export/cvlustre/vulcan/GBT/problem-data). Data that have various problems are placed in this directory. We propose that the directory be flat and only include sub-directories for each distinct session.
6. GB (/home/gbtlogs) to CV (/export/cvlustre/vulcan/GBT/new-logs). Data here are from about the past three months and include various instrumental data logs. We propose that the directory be flat and only include sub-directories for each instrument (e.g., Weather-Weather2-weather2).
7. GB (/home/archive/gbtlogs) to CV (/export/cvlustre/vulcan/GBT/gbtlogs). The archived instrumental data logs are stored here. We propose to maintain the current directory structure that consists of organizing data by years and months.

Currently, there are 15 sessions that are duplicated in /home/archive/science-data. For example, AGBT07C_067.01 through AGBT07C_067.15 appear in two places (tape-0024 and tape-0025). We propose that during the restructuring that when a duplicate session is detected that these duplicates be placed in problem-data with a trailing character string “dupe1” for the first, “dupe2” for the second and so on (e.g., AGBT07C_067.01dupe1, AGBT07C_067.01dupe1, etc.).

Other ancillary data include observing logs, operator logs, and weather forecast data. Since these data exist in databases we propose that they be mirrored to the archive locations in GB and CV on time scales of less than 24 hours. These are indicated by the dotted lines with open arrows. The program used to store operator logs is third-party software and will require some investigation as to whether these data can be mirrored or not. The main goal here, however, is to add this information to the archive. No attempt will be made to access the information in these databases from the AAT at this time, but this will be left to a future project.

3. GBT Meta-data: Historical Context

After each scan the M&C system writes a number of device specific FITS files to disk, including one that contains information that is intended to be indicative of the observer’s intent (i.e., the meta-data) and is written before the scan starts. The meta-data have evolved significantly over the GBT’s lifetime. For historical reasons this FITS file is called the GO

FITS file, and it is usually located in a sub-directory called “GO”. Minter & Prestage (2007) discuss the contents of the GO FITS file, including a history of changes. Each GBT FITS file has a primary header consisting of a number of required keywords (e.g., the date of the observation called DATE-OBS) (Prestage & Clark 2004).

Two observer interfaces have been used: GBT Observe (GO), written in Glish (2001-2005), and the ASTRonomer’s Integrated Desktop (ASTRID), written in Python (2005-present). The GO or ASTRID programs, however, are not necessary to observe. Glish or Python scripts can be used to observe.¹ The evolution of the GO FITS files are, in part, linked to these different observing methods. For example, during early science the GO FITS files appear in a sub-directory called “Glish”. These observations were probably performed using Glish scripts or maybe an early version of GO. Below is an example from project id 01A19 taken on 3 April 2001. This GO FITS file does not conform to the standards in (Minter & Prestage 2007). The contents of these early versions of the GO FITS file are sparse and even the date of the observation is not included.

```
SIMPLE =          T / file does conform to FITS standard
BITPIX =          8 / number of bits per data pixel
NAXIS  =          0 / number of data axes
EXTEND  =          T / FITS dataset may contain extensions
COMMENT  FITS (Flexible Image Transport System) format defined in Astronomy and
COMMENT  Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.
COMMENT  Contact the NASA Science Office of Standards and Technology for the
COMMENT  FITS Definition document #100 and other FITS information.
SCAN    = '15020  ' / integer scan identifier
PROJID  = '01A19  ' / string project identifier
SOURCE  = 'Venus   ' / string source identifier
SCANID  = 'test   ' / string scan identifier
COMMENT  Green Bank Telescope
OBSERVER= 'Campbell' / Name of observer
PROCNAME= 'track   ' / Name of observing procedure
PROCTYPE= 'simple   ' / Observing procedure class
PROCSEQN=          1 / Scan sequence number
PROCSIZE=          1 / Number of scans in procedure
RADECSYS= 'J2000  ' / Coordinate Mode
RAJ2000 = 2.442208333333333E+01 / Commanded J2000 Right Ascension in Degrees
DECJ2000= 3.315983333333333E+01 / Commanded J2000 Declination in Degrees
EQUINOX  = 2.000000000000000E+03 / Equinox of selected coordinate reference frame
END
```

¹Other user interfaces also exist; for example, the Control Library for Engineers and Operators (CLEO) which is, obviously, not intended for astronomers.

The GO FITS file below is a more typical example that does conform to Minter & Prestage (2007) taken in 2004. The number of keywords is still rather limited. For example, there is no indication of the receiver or the back-end that were used. There is some frequency information but it is incomplete. The Spectrometer was configured with 8 spectral windows but none of this information is available in the GO FITS file, but it can, of course, be extracted from other FITS files.

```

SIMPLE =                T / file does conform to FITS standard
BITPIX =                8 / number of bits per data pixel
NAXIS  =                0 / number of data axes
EXTEND  =                T / FITS dataset may contain extensions
COMMENT  FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT  and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
ORIGIN  = 'NRAO Green Bank' /
INSTRUME= 'GO          ' / device or program of origin
GBTCVER= '4.3         ' / telescope control software release
FITSVER  = '1.2        ' / FITS definition version for this device
DATEBLD  = 'Fri Jun 11 13:47:14 UTC 2004' / time program was modified
SIMULATE=                0 / Is the instrument in simulate mode?
DATE-OBS= '2004-06-26T12:47:40' / Manager parameter startTime
TIMESYS  = 'UTC        ' / time scale specification for DATE-OBS
TELESCOP= 'NRAO_GBT'    / Green Bank Telescope (Robert C. Byrd 100m)
OBJECT   = 'S206        ' / Manager parameter source
PROJID   = 'AGBT02A_017_01' / Manager parameter projectId
OBSID    = 'test        ' / Manager parameter scanId
SCAN     =                846 / Manager parameter scanNumber
OBSERVER= 'Balser      ' / Name of observer
PROCNAME= 'OffOn       ' / Name of observing procedure
PROCTYPE= 'SIMPLE      ' / Observing procedure class
PROCSEQN=                2 / Scan sequence number
PROCSIZE=                2 / Number of scans in procedure
OBSTYPE  = 'LINE       ' / Mode of Observation
SWSTATE  = 'PSWITCHON'  / Type of switching
SWCHSIG  = 'TPWCAL    ' / Switching Signals Mode?
LASTON   =                846 / Last On for pos switching
LASTOFF  =                845 / Last Off for pos switching
COORDSYS= 'RADEC      ' / co-ordinate system used to command telescope
RADESYS  = 'FK5        ' / Type of equatorial system in use
EQUINOX  =                2000. / Equinox of selected coordinate reference frame
RA       =                60.816125 / Commanded Major Axis Value in Degrees
DEC      =                51.315 / Commanded Minor Axis Value in Degrees
VELOCITY=                -25400. / Line velocity in rest frame in m/s
VELDEF   = 'VRAD-LSR'   / Velocity definition and reference frame
RESTFRQ  =                9183000000. / line rest frequency in Hz
END

```

During the transition from GO to ASTRID additional information was included under the HISTORY keyword. This is not discussed in Minter & Prestage (2007) and not well documented elsewhere to our knowledge. The additional information includes configuration and catalog data. The information is listed after “Configuration keywords:” and “Catalog attributes:” for the configuration and catalog data, respectively.

Unfortunately, the configuration and catalog information in these HISTORY keywords may not accurately represent user intent. Based on experience the configuration information appears to be correct much of the time. For example, the rest frequency data in the HISTORY keywords is identical to the values in the main keywords.² But there are some known issues. For example, when using the Zpectrometer the configuration information is incorrect where obstype is “Continuum” and backend is “DCR” (see below).

```

SIMPLE =          T / conforms to FITS standard
BITPIX =          8 / array data type
NAXIS  =          0 / number of array dimensions
EXTEND =          T
ORIGIN = 'NRAO Green Bank'
TIMESYS = 'UTC      ' / time scale specification for DATE-OBS
OBJECT  = 'SMMJ02400-0134' / Manager parameter source
SCAN    =          37 / integer scan identifier
LASTON  =          37 / Last On for pos switching
OBSERVER= 'Andrew Harris' / Name of observer
DATE-OBS= '2008-02-23T00:53:53' / Manager parameter startTime
DATEBLD = '2008-02-06T19:12:40' / time program was released
TELESCOP= 'NRAO_GBT' / Green Bank Telescope (Robert C. Byrd 100m)
GBTCVER= '8.1      ' / telescope control software release
INSTRUME= 'Turtle  ' / device or program of origin
PROJID  = 'AGBT06C_033_30' / Manager parameter projectId
SIMULATE=          0 / Is the instrument in simulate mode
OBSID   = 'unknown ' / Manager parameter scanId
LASTOFF =          0 / Last Off for pos switching
FITSVER = '2.6      ' / FITS definition version for this device
BLOCK   = 'zpec07-obs-fastnod-duo'
COORDSYS= 'RADEC   '
PROCNAME= 'Track    '
PROCTYPE= 'SIMPLE   '
PROCSIZE=          1
PROCSEQN=          1

```

²The rest frequencies in the main keywords do not include the “deltafreq” offset that can be specified in the configuration. This is a reasonable choice since these offset are usually applied with the intent to be offset from some spectral line.


```

VELOCITY=                0.0
OBSTYPE = 'CONTINUUM'
SWCHSIG= 'BEAMSW  '
SWSTATE = 'BSWITCH '
VELDEF  = 'VRAD-TOP'
RESTFRQ =                32000000000.0
SKYFREQ =                32000000000.0
SUBMOTIN= 'SubNod  '      / Subreflector control object
SUBSTRTY=                -90.0 / Scan start position on Y-axis in mm
SUBOFFSY=                180.0 / Scan offset on Y-axis in mm
SUBNODLN= '5.0      '      / Half-cycle time
SUBNUNIT= 'seconds '      / Half-cycle time units
SUBXT1  = -0.07803121248499399 / beam 1 x tilt in degrees
SUBYT1  =                0.0 / beam 1 y tilt in degrees
SUBZT1  =                -0.0 / beam 1 z tilt in degrees
SUBXT2  =  0.07803121248499399 / beam 2 x tilt in degrees
SUBYT2  =                0.0 / beam 2 y tilt in degrees
SUBZT2  =                -0.0 / beam 2 z tilt in degrees
RADESYS = 'FK5      '
EQUINOX =                2000.0
RA       =                39.99125
DEC      =   -1.579166666666667
HISTORY Configuration keywords:
HISTORY vdef = "Radio"
HISTORY swtype = "bsw"
HISTORY deltafreq = 0
HISTORY pol = "Circular"
HISTORY swper = 0.1
HISTORY vframe = "topo"
HISTORY swmode = "sp"
HISTORY backend = 'DCR'
HISTORY swfreq = 0,0
HISTORY tint = 0.1
HISTORY beam = 'B1'
HISTORY vhigh = 0.0
HISTORY bandwidth = 320
HISTORY nwin = 1
HISTORY noisecal = "lo"
HISTORY receiver = 'Rcvr26_40'
HISTORY restfreq = 32000
HISTORY obstype = 'Continuum'
HISTORY vlow = 0.0
HISTORY Catalog attributes:
HISTORY source : SMMJ02400-0134
HISTORY location : J2000 @ (02:39:57.89, -01:34:45.00)
END

```

Most of the problems detected in the HISTORY file, however, are with the catalog information. For example, the GO FITS file below was generated in ASTRID from a scheduling block that was looping through a list of target sources from a catalog file. The scheduling block checks whether or not to perform a *Peak* on the pointing calibrator 1713-2658 from the standard pointing catalog, called “pointing”, every 2 hours. When the target was G30.0-22.0 it was time to point. The source name from the keyword OBJECT is “1713-2658”, but the catalog HISTORY information has G30.0-22.0 for both the name and location.

```

SIMPLE = T / conforms to FITS standard
BITPIX = 8 / array data type
NAXIS = 0 / number of array dimensions
EXTEND = T
ORIGIN = 'NRAO Green Bank'
TIMESYS = 'UTC' / time scale specification for DATE-OBS
OBJECT = '1713-2658' / Manager parameter source
SCAN = 1 / integer scan identifier
LASTON = 1 / Last On for pos switching
OBSERVER= 'Dana Balser' / Name of observer
DATE-OBS= '2011-10-14T22:20:02' / Manager parameter startTime
DATEBLD = '2011-05-22T05:35:07' / time program was released
TELESCOP= 'NRAO_GBT' / Green Bank Telescope (Robert C. Byrd 100m)
GBTCVER= '11.3' / telescope control software release
INSTRUME= 'Turtle' / device or program of origin
PROJID = 'AGBT11B_010_01' / Manager parameter projectId
SIMULATE= 0 / Is the instrument in simulate mode
OBSID = 'unknown' / Manager parameter scanId
LASTOFF = 0 / Last Off for pos switching
FITSVER = '2.7' / FITS definition version for this device
BLOCK = 'teLow'
SWSTATE = 'NONE'
COORDSYS= 'RADEC'
PROCNAME= 'Peak'
PROCTYPE= 'POINTING'
PROCSIZE= 4
PROCSEQN= 1
VELOCITY= 0.0
OBSTYPE = 'CONTINUUM'
SWTCHSIG= 'TPWCAL'
VELDEF = 'VRAD-TOP'
RESTFRQ = 8665000000.0
RESTFRQ1= 8665000000.0
SKYFREQ = 8665000000.0
SUBMOTIN= 'SubFocus' / Subreflector control object
SUBSTRTY= -240.0 / Scan start position on Y-axis in mm

```

```

SUBOFFSY=                480.0 / Scan offset on Y-axis in mm
REFBEAM = '-1          '
RADESYS = 'FK5          '
EQUINOX =                2000.0
RA      =                258.380315
HISTORY Configuration keywords:
HISTORY vdef = "Radio"
HISTORY swtype = "none"
HISTORY deltafreq = 0
HISTORY pol = "Circular"
HISTORY swper = 0.1
HISTORY vframe = "topo"
HISTORY swmode = "tp"
HISTORY backend = 'DCR'
HISTORY swfreq = 0,0
HISTORY tint = 0.1
HISTORY beam = 'B1'
HISTORY vhigh = 0.0
HISTORY bandwidth = 320
HISTORY nwin = 1
HISTORY noisecal = "lo"
HISTORY receiver = 'Rcvr8_10'
HISTORY restfreq = 8665
HISTORY obstype = 'Continuum'
HISTORY vlow = 0.0
HISTORY Catalog attributes:
HISTORY source : G30.0-22.0
HISTORY vel : 0.0
HISTORY location : Galactic @ (29:12:54.00, -22:00:00.00)
END

```

There are instruments (mostly back-ends) that are not fully integrated into the M&C system. Many of these are user supplied back-ends but they also include NRAO instruments. In some of these cases the information in the GO FITS file is limited or incorrect (e.g., Zpectrometer). Sometimes back-end data are not stored locally and are therefore not part of the GBT archive. For example, VLBI and GUPPI observations are typically not stored in the standard areas for technical reasons. Below is a GO FITS file using the pulsar back-end GUPPI.

```

SIMPLE =                T / conforms to FITS standard
BITPIX =                8 / array data type
NAXIS  =                0 / number of array dimensions
EXTEND =                T
ORIGIN = 'NRAO Green Bank'

```

```

TIMESYS = 'UTC'           / time scale specification for DATE-OBS
OBJECT  = 'J2010-1323'    / Manager parameter source
SCAN    =                 30 / integer scan identifier
LASTON  =                 30 / Last On for pos switching
OBSERVER= 'Paul Demorest' / Name of observer
DATE-OBS= '2011-11-28T16:44:22' / Manager parameter startTime
DATEBLD = '2011-05-22T05:35:07' / time program was released
TELESCOP= 'NRAO_GBT'      / Green Bank Telescope (Robert C. Byrd 100m)
GBTCVER= '11.3'          / telescope control software release
INSTRUME= 'Turtle'        / device or program of origin
PROJID  = 'AGBT09B_041_55' / Manager parameter projectId
SIMULATE=                 0 / Is the instrument in simulate mode
OBSID   = 'Map'           / Manager parameter scanId
LASTOFF =                 0 / Last Off for pos switching
FITSVER = '2.7'           / FITS definition version for this device
BLOCK   = 'NEW_Observe_GUPPI2'
COORDSYS= 'RADEC'
PROCNAME= 'Track'
PROCTYPE= 'SIMPLE'
PROCSIZE=                 1
PROCSEQN=                 1
VELOCITY=                 0.0
OBSTYPE  = 'UNKNOWN'
SWCHSIG  = 'TPNOCAL'
SWSTATE  = 'NONE'
VELDEF   = 'VRAD-TOP'
RESTFRQ  =                 820000000.0
RESTFRQ1=                 820000000.0
SKYFREQ  =                 820000000.0
RADESYS  = 'FK5'
EQUINOX  =                 2000.0
RA        =                 302.6912916666667
DEC       =                 -13.39888888888889
HISTORY Configuration keywords:
HISTORY pol = 'Linear'
HISTORY vframe = 'topo'
HISTORY backend = 'GUPPI'
HISTORY swfreq = 0.0, 0.0
HISTORY tint = 10.24e-6
HISTORY vhigh = 0
HISTORY guppi.fold_bins = 2048
HISTORY guppi.polnmode = 'full_stokes'
HISTORY nwin = 1
HISTORY restfreq = 820.0
HISTORY guppi.fold_dumptime = 15.0
HISTORY guppi.fold_parfile = '/users/pdemores/tzpar/J2010-1323.par'

```

```

HISTORY vdef = 'Radio'
HISTORY swtype = 'none'
HISTORY deltafreq = 0
HISTORY guppi.outbits = 8
HISTORY swper = 0.04
HISTORY swmode = 'tp_nocal'
HISTORY guppi.scale = 0.125
HISTORY guppi.numchan = 128
HISTORY vlow = 0
HISTORY ifbw = 0
HISTORY bandwidth = 200
HISTORY noisecal = 'off'
HISTORY guppi.obsmode = 'coherent_fold'
HISTORY receiver = 'Rcvr_800'
HISTORY obstype = 'Pulsar'
HISTORY guppi.datadisk = 'data1'
HISTORY Catalog attributes:
HISTORY alias : 2010.0
HISTORY stop_lst : 16:30:00
HISTORY location : J2000 @ (20:10:45.91, -13:23:56.00)
HISTORY source : J2010-1323
HISTORY start_lst : 15:50:00
HISTORY jname : J2010-1323
END

```

4. GBT Meta-data: Future

Currently, GBT meta-data are contained across FITS files, and meta-data that specify user intent are included in GO FITS files. The GO FITS file is written before the scan starts but other FITS files (e.g., back-end) are written when the scans finishes. We therefore recommend that the GBT meta-data be generated after the scan finishes on a timescale of less than 24 hours. The meta-data should consist of a comma separated variable (CSV) file and primarily contain information about user intent that can be used by archive access tools and potentially a pipeline.

In this section we discuss changes to the GO FITS file that include adding a few keywords and a binary table to better capture user intent. We then discuss meta-data that are derived from the various FITS files. We also recommend how best to extract this information from legacy data.

4.1. GO FITS File Keywords

In Table 1 we list the proposed keywords that should exist in future GO FITS files. Listed are the keyword name, the type, the current status, and comments. The status indicates if the keyword currently exists or not. This is not a complete list of keywords since the GO FITS files have other consumers. Below we discuss each keyword in detail, including how this keyword should be extracted from legacy GBT data.

4.1.1. *PROJID*

The PROJID keyword in the GO FITS file is the Scan Coordinator manager parameter “ProjectId” and is a primary header keyword (Minter & Prestage 2007). This appears to be the sub-directory name for all data, including early science. For the GO observer interface this keyword could be arbitrarily set by the user, whereas ASTRID restricts the possible values using a menu list. A PROJID keyword value exists for all scans in the science-data directory.

For most scans the PROJID keyword consists of the proposal id preceded with an “A” for astronomy and trailed by the session number. For example, AGBT11B.010.03 where the proposal id is GBT/11B-010. This convention has been reasonably consistent over time but there are anomalies, especially during early science. Moreover, for VLBI observations the legacy proposal code is used (e.g., ABL175.02) which can be traced back to the proposal. Therefore, the proposal id can be parsed from the PROJID keyword in most cases. This is important for connecting the investigators listed for this proposal to the data when accessing proprietary data. But there are PROJID keywords that are not associated with any NRAO proposal. For example, cooperative agreements between the NRAO and other facilities (e.g., Fermi) allow proposals to be reviewed externally. In these cases the PROJID keyword is usually representative of this fact (e.g., AGLST031250.22 for a Fermi proposal).

We recommend that the PROJID keyword value be directly extracted from all GO FITS files.

4.1.2. *OBJECT*

The OBJECT keyword in the GO FITS file is the Scan Coordinator manager parameter “source” and is a primary header keyword (Minter & Prestage 2007). This keyword is set for

Table 1. GBT GO FITS File Keywords

Keyword	Type	Status	Comment
PROJID	string	✓	Project ID (e.g., AGBT11A_043_12).
OBJECT	string	✓	Target source name (e.g., 3C286).
OBSERVER	string	✓	Observer as specified by Astrid (not related to proposal).
DATE-OBS	string	✓	Date-time with units of yyyy-mm-ddThh:mm:ss.
SCAN	string	✓	The scan number (e.g., 42).
PROCNAME	string	✓	Observing procedure name (e.g., OffOn).
PROCTYPE	string	✓	Observing procedure type (e.g., MAP).
OBSTYPE	string	✓	Observing type (e.g., LINE).
OBSID	string	✓	Observing ID for user input.
VELDEF	string	✓	Velocity definition (e.g., VRAD-LSR).
VELOCITY	float	✓	Source velocity in units of m/s.
RESTFn ^a	float	✓	Rest frequency in units of Hz.
BWn ^a	float	×	Band width in units of Hz.
NCHANn ^a	float	×	Number of channels.
POLN	string	×	Polarization products (e.g., Linear-Dual).

^aThe integer n is the count (e.g., RESTF1, RESTF2, etc.)

all but 149 scans in the science-data directory.³ These scans were taken in 2003-2004 using the GO interface. Presumably the user left this field blank which is allowed. As discussed above the catalog information in the GO FITS file HISTORY keywords are not reliable, but it appears the OBJECT keyword is providing the correct user intended source name. There are exceptions, however, where user procedures are defined that do not properly set the OBJECT keyword.

We recommend that the OBJECT keyword value be directly extracted from all GO FITS files.

4.1.3. *OBSERVER*

The OBSERVER keyword is a shared keyword that has an agreed definition and interpretation but is not a primary header keyword (Prestage & Clark 2004). OBSERVER is a string giving the observer’s name. It appears to exist only in the GO FITS file. For the GO observer interface this keyword could be arbitrarily set by the user. ASTRID restricts the possible values from a finite set of valid user names from a menu list. An OBSERVER keyword value exists for all but 7 scans in the science-data directory. These are associated with PROJID AGBT04B.026_01 taken on 2004-07-24 with the GO interface.

We recommend that the OBSERVER keyword value be directly extracted from all GO FITS files.

4.1.4. *DATE-OBS*

The DATE-OBS keyword in the GO FITS file is the manager parameter “startTime” and is a primary header keyword (Prestage & Clark 2004). During early science, however, there are some projects that presumably used Glish scripts to observer where the DATE-OBS keyword does not exist (see §3).

We recommend that the DATE-OBS keyword value be directly extracted from all GO FITS files. For early science data that do not include the DATE-OBS keyword it should be determined using the GO FITS file name which includes the date/time. Some DATE-OBS keyword values have a year of 1858. This is should be investigated and fixed if still a problem.

³Here we do not distinguish between scans where the keyword does not exist in the GO FITS file or when the keyword does exist but is blank (i.e., “”) or a space (i.e., “ ”).

For legacy data these should be detected and fixed by using the GO FITS file name.

4.1.5. SCAN

The SCAN keyword in the GO FITS file is a primary header keyword as well as a pre-existing FITS keyword (Prestage & Clark 2004). The SCAN keyword exists for all scans in the science-data directory.

We recommend that the SCAN keyword value be directly extracted from all GO FITS files.

4.1.6. PROCNAME

The PROCNAME keyword in the GO FITS file is the procedure name used in either the GO or ASTRID observing programs. In most cases these are well defined procedures that have been sanctioned by the NRAO, but the users can define their own procedures. There are a total of 30 different distinct values of PROCNAME. The PROCNAME keyword exists for all but 5 scans in the science-data directory. All of these scans were taken using the Rcvr_PAR (MUSTANG) receiver running a scheduling block called “findbestbiases”. These scheduling blocks are listed only 45 times and do not generate any scientifically relevant data (Mason, priv. Comm.). Users can create their own procedures but there is *no* mechanism to force them to change the name. Therefore, there is no guarantee that a standard procedure name has indeed been used.

We recommend that the PROCNAME keyword value be directly extracted from all GO FITS files. Scans that do not contain this keyword or have blank values should be set to “Unknown”. For legacy data the ASTRID logs should be used to identify when users are using a user-defined procedures that has a standard procedure name. For these cases the procedure name should be set to “PROCNAME_OBSERVER” (e.g., Nod.Harris).

4.1.7. PROCTYPE

The PROCTYPE keyword in the GO FITS file describes the category of procedure used for the observation. Only 5 scans have no values for PROCTYPE in the science-data directory which are identical to the 5 missing values for PROCNAME. The following values are used: POINTING, SIMPLE, CALIBRATION, MAP, OOFMAP, UNKNOWN, AND “”.

We recommend that the PROCTYPE keyword value be directly extracted from all GO FITS files. Scans that do not contain this keyword or have blank values should be set to “UNKNOWN”.

4.1.8. OBSTYPE

The OBSTYPE keyword in the GO FITS file provides information about the type of observing that was performed. An OBSTYPE keyword value exists in all scans in the science-data directory. The values found include CONTINUUM, LINE, Spectroscopy, VLBI, RADAR, PULSAR, PTIMING, PSEARCH, PMONITOR, and UNKNOWN. The GO interface does not include “Spectroscopy”, whereas ASTRID includes both “LINE” and “Spectroscopy”. Currently, ASTRID sets this keyword based on the configuration parameter “obstype”. There are two different values for spectral line in ASTRID since early versions of the software used the configuration value (“Spectroscopy”), but this is now converted to “LINE”.

We recommend that the OBSTYPE keyword value be directly extracted from all GO FITS files. Scans that do not contain this keyword or have blank values should be set to “UNKNOWN”. All scans that have a value of “Spectroscopy” should be set to “LINE” since these appear to be the same type.

4.1.9. OBSID

The OBSID keyword in the GO FITS file is the manager parameter “scanId” and is a primary header keyword (Prestage & Clark 2004). The intent is that this string can be set by the user to specify some aspect of this scan. In principle it could be used by data analysis programs or a pipeline. For example, the KPFA pipeline does use this keyword.

We recommend that the OBSID keyword value be directly extracted from all GO FITS files.

4.1.10. Velocity (VELDEF, VELOCITY)

The VELDEF and VELOCITY keyword values are set in the GO FITS file for 93.2% of the scans in the science-data directory. They are always set when OBSTYPE is “LINE” or “Spectroscopy” and they are often set for other observing types (CONTINUUM, PULSAR).

The following VELDEF keyword values exist in the science-data directory: VRAD-TOP, VRAD-LSR, VRAD-BAR, VRAD-HEL, VRAD-GAL, VRAD-LSD, VOPT-TOP, VOPT-LSR, VOPT-BAR, VOPT-HEL, VOPT-LSD, VELO-TOP, and VELO-LSR. For early science data there are scans where VELDEF and VELOCITY are not included. Currently, VELDEF is extracted from the ASTRID configuration, whereas the VELOCITY is derived from the the SOUVEL binary table of the LOA1 FITS file (see Brandt & Prestage 2001), which is written before the scan starts. The LO1A FITS files do not always exists for early science.

We recommend that the VELDEF and VELOCITY keyword values be directly extracted from all GO FITS files. For early science data set these keywords to “”.

4.1.11. Frequency (*RESTFn*, *BWn*, *NCHANn*)

The RESTFRQn keyword was added to the GO FITS file on 14 May 2010 (Minter & Prestage 2007) and consists of the user supplied rest frequencies. These are taken from the configuration and should be identical to the configuration HISTORY keyword frequencies. There is also a single keyword SKYFREQ which corresponds to the intended sky frequency. It is not clear how this value is calculated. For legacy GBT data there is typically the keyword RESTFREQ in the GO FITS file when the OBSTYPE keyword value is “LINE” that corresponds to the line rest frequency of the first spectral window. For OBSTYPE keyword values that are not “LINE” there is typically the keyword SKYFREQ in the GO FITS file that corresponds to the intended sky center frequency. Currently RESTFREQ, RESTFRQ1, and SKYFREQ exist for all values of OBSTYPE.

We recommend adding the keywords BWn and NCHANn to the GO FITS file for all values of OBSTYPE. We suggest changing RESTFRQn to RESTFn since the new back-end VEGAS will have more than 9 spectral windows and a main FITS header keyword can only have 8 characters. For each of the n spectral windows BWn and NCHANn are the corresponding bandwidths and number of channels, respectively. For continuum observations (e.g., DCR) BWn should be taken from the IF FITS file, whereas for line data both BWn and NCHANn should be taken from the back-end FITS file.

4.1.12. *POLN*

We recommend adding the keyword `POLN`⁴ to the GO FITS files with possible values listed in Table 2. Basically there are two parts to the keyword. The first part indicates whether the products are linear or circular, and the second part is the number of products (single, dual, or full). We suggest using the IF FITS file to derive the first part. Back-end specific parameters usually specify if full polarization is desired (e.g., “spect.crosspol” for the Spectrometer). The assumption is that you will get dual polarization by default and need to specify adding the cross products or full polarization. This is usually a good assumption for the GBT but there are exceptions (e.g., Ka-band receiver which currently only has single polarizations). Also, the Ka-band receiver used to be native circular but now is linear. This is currently incorrect in the IF FITS file and should be fixed.

4.2. GO FITS Binary Table

We propose that the procedure parameter information be listed in a binary table called “OBS_PROC”. The binary table will have dimension $n \times 1$ where each column is the procedure parameter and the first row is their corresponding value. Below is an example of the header for the OnOff procedure. For completeness we have added the main header keywords involving the procedure. We have also added `PROC NRAO` as a true/false flag indicating if the procedure is a standard NRAO procedure.

⁴We have not used `POLARIZE` since a keyword with this name is used elsewhere (Bob Garwood, priv. comm.).

Table 2. GBT Keyword `POLN`

Keyword	Value	Comment
Linear-Single	XX or YY	
Linear-Dual	XX and YY	
Linear-Full	XX, YY, XY, and YX	
Circular-Single	LL or RR	
Circular-Dual	LL and RR	
Circular-Full	LL, RR, LR, and RL	

One complication is that ASTRID procedures contain objects. The two main objects are the location object and the offset object. These objects are broken up into individual keywords below with a the preface “LO” and “OO” for location object and offset object, respectively. The location object contains the coordinate mode and the longitude and latitude. The offset object has these same parameters for an offset position plus a flag for whether or not to apply a cosv correction. The header should also include the Python code of the procedure as comments at the end (not included below). This will provide detailed information about the procedure.

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                8 / 8-bit bytes
NAXIS   =                2 / 2-dimensional binary table
NAXIS1  =               72 / width of table in bytes
NAXIS2  =                1 / number of rows in table
PCOUNT  =                0 / size of special data area
GCOUNT  =                1 / one data group (required keyword)
TFIELDS =                9 / number of fields in each row
PROCNAME= 'OffOn   '         / procedure name
PROCTYPE= 'SIMPLE '         / procedure type
PROCSEQN=                1 / procedure sequence number
PROCSIZE=                2 / procedure size
PROCNRAO= 'True   '         / standard NRAO procedure
TTYPER1 = 'LO_COORD'        / label 1: Location Object location
TFORM1  = '1D   '         / data format of field: 8-byte STRING
TUNIT1  = '      '         / physical unit of field
TTYPER2 = 'LO_LONG '        / label 2: Location Object location
TFORM2  = '1D   '         / data format of field: 8-byte DOUBLE
TUNIT2  = 'deg   '         / physical unit of field
TTYPER3 = 'LO_LAT  '        / label 3: Location Object location
TFORM3  = '1D   '         / data format of field: 8-byte DOUBLE
TUNIT3  = 'deg   '         / physical unit of field
TTYPER4 = 'OO_COORD'        / label 4: Offset Object referenceOffset
TFORM4  = '1D   '         / data format of field: 8-byte STRING
TUNIT4  = '      '         / physical unit of field
TTYPER5 = 'OO_LONG '        / label 5: Offset Object referenceOffset
TFORM5  = '1D   '         / data format of field: 8-byte DOUBLE
TUNIT5  = 'deg   '         / physical unit of field
TTYPER6 = 'OO_LAT  '        / label 6: Offset Object referenceOffset
TFORM6  = '1D   '         / data format of field: 8-byte DOUBLE
TUNIT6  = 'deg   '         / physical unit of field
TTYPER7 = 'OO_COSV '        / label 7: Offset Object referenceOffset
TFORM7  = '1D   '         / data format of field: 8-byte STRING
TUNIT7  = '      '         / physical unit of field
TTYPER8 = 'SCANDURATION'    / label 8: scanDuration
```

```

TFORM8 = '1D'          / data format of field: 8-byte DOUBLE
TUNIT8 = 'sec'         / physical unit of field
TTYPE9 = 'BEAMNAME'    / label 9: beamName
TFORM9 = '1D'          / data format of field: 8-byte DOUBLE
TUNIT9 = '             ' / physical unit of field
EXTNAME = 'OBS_PROC'    / name of this binary table extension
END

```

4.3. GBT Derived Meta-data

Table 3 summarizes meta-data that should be derived from data in various FITS files. Each of these is discussed in detail below.

4.3.1. RECEIVER

A FITS keyword called RECEIVER is defined (Prestage & Clark 2004) but appears to only exist in the Receiver FITS file. This information does appear in the configuration HISTORY keywords but it may be incorrect. We therefore recommend that the IF FITS file be used to determine the receiver name. Only one receiver can be connected for a given scan and so this should be unique. If the IF FITS file is not available then the configuration HISTORY keyword can be used. If the receiver cannot be determined then RECEIVER should be set to “UNKNOWN”.

4.3.2. BACKEND

A FITS keyword called BACKEND is defined (Prestage & Clark 2004) but appears to only exist in the back-end FITS file (e.g., Spectrometer). This information does appear in the configuration HISTORY keywords but it may be incorrect. We therefore recommend that the IF FITS file be used to determine the back-end name. Since many back-ends can be connected but not used the ScanLog.fits file is necessary to determine which back-ends were taking data. If the IF FITS file is not available the configuration HISTORY keyword can be used.

For some observations multiple back-ends are used simultaneously. These should all be included and separated by an appropriate character (e.g., colon).

4.3.3. Location (RAJ2000, DECJ2000, OBSAZ, OBSEL)

The GBT legacy data GO FITS files contain positional information (see Minter & Prestage 2007)⁵. These positions are *not* actual encoder positions but they specify the target position in the coordinates in which they were specified by the user (e.g., J2000, Galactic, etc.). For most procedures there is a location field that indicates the target position (i.e., the user supplied coordinates to the observing procedure which is typically the source position or the center position of a map). For example, an OffOn procedure consists of two scans where the commanded positions are different. The target position is the On position.

For the GO observing interface the target position is used for all scans for a given procedure. So an OffOn scan will list the same position for both the Off and On scans. But for ASTRID the Off and On scans will be different whereby the On scan lists the target position and the Off scan lists the target position plus the specified offset.

We recommend using the positional information in the GO FITS file to derive the J2000 RA and Dec positions (RAJ2000 and DECJ2000). These positions should *not* include any offsets but be indicative of the target location or center of a map. That is, they should be the location indicated in the ASTRID procedure. For objects that require an ephemeris (e.g., solar system objects) mean values of the ephemeris positions should be used.

We also recommend that the expected telescope (Az, El) position, called OBSAZ and OBSEL, be derived using the RAJ2000 and DECJ2000 coordinates and the DATE-OBS keyword to estimate the telescope azimuth and elevation. The main reason for adding these coordinates is to allow users to search on elevation since the weather, important for higher frequency observations, is elevation dependent.

We propose to *not* include encoder positions since they are not necessarily representative of user intent. One could use the encoder positions, however, to provide a sanity check of the target location. But given the flexibility of the GBT we suspect there will always be some issues with meta-data accuracy and our goal here is to derive user intent most of the time. There are some situations where we know the GO FITS file positions are incorrect. When user defined coordinates are specified there is no positional information available and the target location is unknown. Also, observations taken when the telescope drives are turned off may produce incorrect target positions. These situations should be fixed if possible by providing this information to the GO FITS files.

⁵The RA keyword in the GO FITS file was being written in hours instead of degrees for a period of time. See https://safe.nrao.edu/wiki/bin/view/GB/Software/ObservingAPIReleaseNotes_7.1.

4.3.4. *VERSION*

Since the GBT meta-data have evolved considerably over time we recommend deriving a version number (*VERSION*) based on the *DATE-OBS* keyword.. There are several natural break points during this evolution and we suggest using these to define different versions. We suggest the following. *VERSION* equal to 1.0 for all early science since these data were generated either by using scripts or early versions of the GO interface where the meta-data were very limited. *VERSION* equal to 2.0 for data generated by the GO interface after early science. *VERSION* equal to 3.0 for data generated by ASTRID. These data included *HISTORY* keywords that, while sometimes incorrect, provided more complete frequency information. Lastly, *VERSION* equal to 4.0 for data taken after the current proposal has been implemented with much more complete meta-data.

There is a primary FITS header keyword called *FITSVER* that specifies the version of each device FITS file. Since we are deriving the meta-data from more than just one of these FITS files, however, there is no single *FITSVER* value that corresponds to *VERSION*; hence we suggest deriving a new version for the meta-data.

4.3.5. *PROPID*

PROPID is the proposal id that in most cases can be directly extracted from the GO FITS file keyword *PROJID* keyword. *PROJID* keyword values like AGBT11B_010_02 will have corresponding proposal id's of GBT/11B-010.

4.3.6. *BAND*

BAND is the observing band using the standard radio nomenclature. The *BAND* can be derived from the receiver name using *RECEIVER*.

4.3.7. *FILENAME*

The file name associated with the scan is a string given in the format *YYYY_MM_DD_HH:MM:SS.fits*. This is the file name of the FITS files from which the primary meta-data have been derived. This is not the pointer for data retrieval that would be used by archive access tools or pipelines.

4.3.8. *FILESIZE*

The file size of *FILENAME* in kilo bytes. For example, it could be computed via the Python method `os.path.getsize(filename)/1024.0`.

4.3.9. *MSGFLAG*

When the program that generates the meta-data detect problems with the meta-data they are recorded here with a message.

4.3.10. *BADDATA*

BADDATA is a flag that indicates if these data are good (0) or bad (1). *BADDATA* will be set to 0 by default. In the future a mechanism could be put in place that would allow users to mark data as bad.

4.4. Pulsar Data

Pulsar data are typically either not saved or are archived separately and thus not found in the science-data directory. The pulsar community, moreover, has developed their own FITS files (PSRFITS) for the specific storage of pulsar data. PSRFITS files do include meta-data. We therefore propose extracting meta-data from both legacy and future PSRFITS files for pulsar data archived by the NRAO. We recommend mapping all meta-data from the PSRFITS file to the meta-data discussed in §4.1—§4.3.

4.5. Meta-data Generation

The GBT meta-data should be generated after a scan finishes on timescales of less than 24 hours. We recommend using a single CSV formatted file that includes the meta-data discussed in §4.1—§4.4. The code used to generate GBT meta-data should be maintained by the GB software group. The meta-data should reside in the science-data directory (/home/archive/science-data for GB and /export/cvlustre/vulcan/GBT/science-data for CV). An automated mechanism is required with notification of any problems.

Below is an example of a row in the CSV meta-data file that corresponds to a single

scan. A header is included. In this example the information for a single scan is on multiple lines for clarity but in practice each scan should encompass one row. Since there can be more than one value for the rest frequency, bandwidth, and number of channels these are formatted like a Python tuple. The information in the GO FITS file binary table includes procedure parameters and is also formatted as a tuple.

```
PROJID, OBJECT, OBSERVER, DATE-OBS,
SCAN, PROCNAME, PROCTYPE, OBSTYPE, OBSID, VELDEF, VELOCITY,
RESTF,
BW,
NCHAN, POLN,
OBS_PROC,
RECEIVER, BACKEND, RAJ2000, DECJ2000,
OBSAZ, OBSEL, VERSION, PROPID, BAND,
FILENAME, FILESIZE, MSGFLAG, BADDATA

"AGBT11B_010_07", "G032.876-0.423", "Karl Jansky", "2012-01-24T14:28:55",
"42", "OffOn", "SIMPLE", "LINE", "off", "VRAD-LSR", "50600.0",
"9183000000.0,9505000000.0,9812000000.0,8665300000.0",
"500000000.0,500000000.0,500000000.0,500000000.0",
"4096,4096,4096,4096", "Circular-Dual",
"OffOn,SIMPLE,1,2,True,J2000,283.2107083,-0.2426388888,J2000,-1.604,0.0,False,360.0,1"
"Rcvr8_10", "Spectrometer", "283.2107083", "-0.2426388888",
"195.908", "50.194", "3.0", "GBT/11B-010", "X",
"2012_01_24_14:28:55.fits", "8.5", "", "0"
```

5. Archive Access Tool (AAT)

The Archive Access Tool (AAT) is currently utilized by the VLA, EVLA, and VLBA to allow access to various data products that include raw data files, pipeline processed images, operator logs, etc. Tools are available that search on corresponding meta-data from these three instruments. Currently data can be retrieved in various data formats (ASDM, CASA measurement sets, etc.). We propose to include the GBT in the AAT by incorporating GBT legacy and future meta-data as specified in §4 into the AAT.

The mapping of GBT meta-data to the AAT index files are shown in Tables 4-10. Items with checkmarks are currently incorporated into the AAT; items with crosses need to be added. The AAT index files should be generated on timescales of less than 24 hours. An automated mechanism is required with notification of any problems.

We propose only minor changes to the AAT user interface at this time. A checkbox for the GBT should be included under telescope. Three lists should be added to the AAT that allow users to search on RECEIVER, BACKEND, and PROCNAME. Additional modifications can be made later after we get user feedback.

Currently the AAT allows data requests in different formats (e.g., ASDM or CASA measurement sets for the EVLA). We suggest two formats for the GBT: (i) raw FITS files; and (ii) SDFITS file. These should be available as a compressed tar file. The sdfits program should be available in CV and working on 64-bit architecture. The user should be able to specify the different options in the sdfits program (e.g., -backends=acs).

6. Data Quality and Products

The meta-data discussed in §4 provide information about user intent, but they do not tell the user anything about what really happened or anything about the data quality. Such information is useful. For example, assume a user is searching for data toward a given location near the K-band water line. Let us also assume that there are many observations for this particular location and frequency but much of the data were taken under poor weather conditions (maybe before the dynamic scheduling system was in operation). The weather information would be very useful in deciding what data to process. Observing and Operator logs would also benefit any individual attempting to process data.

In general it seems worthwhile to have a summary about what happened. For example, if the observation was aborted, the weather, the rms noise of the spectrum or image, etc. But the best metrics require actually processing data. Such a summary could be generated by the GBT pipeline with products similar to the VLA image archive.

Generating data quality summaries or data products is beyond the scope of this project. But we strongly recommend that future projects should be considered to put such information back into the GBT archive that can be searched and retrieved by the AAT as is currently done with other NRAO telescopes. For example, this could be a future task of the GBT pipeline group.

7. Observing Policies

To ensure the integrity of the meta-data certain observing policies must be met. Below we list changes to current observing policies.

1. In the past there was no restriction that a user observe under the ASTRID project id for which they were being billed time. For example, a user could observe project AGBT11B_032 under the project code AGBT08A_12. So users associated with AGBT11B_032 would not be able to retrieve these data if they were also not an author on proposal AGBT08A_12. It should be confirmed that an observing policy is in place where this never occurs.
2. Currently users can create new observing procedures that have names identical to existing standard NRAO procedures. In the future ASTRID should prohibit users from executing scheduling blocks with these types of procedures.

This document evolved considerably based on discussions with the GB scientific staff, the GB software division, and the GBT pipeline group.

REFERENCES

- Brandt, J. J., & Prestage, R. M. 2007, GBT Software Project Note 6.5
- Brandt, J. J., & Prestage, R. M. 2008, GBT Software Project Note 7.13
- Minter, T., & Prestage, R. 2007, GBT Software Project Note 8.6
- Prestage, R. M. & Clark, M. H. 2004, GBT Software Project Note 4.2

A. Division of Labor

In Table 11 we summarize the division of labor.

Table 3. GBT Derived Meta-data

Keyword	Type	Comment
RECEIVER	string	Receiver (front-end) name (e.g., Rcvr8_10).
BACKEND	string	Detector (back-end) name (e.g., Spectrometer).
RAJ2000	float	J2000 Right Ascension in units of degrees.
DECJ2000	float	J2000 Declination in units of degrees.
OBSAZ	float	Expected Observed Azimuth in units of degrees.
OBSSEL	float	Expected Observed Elevation in units of degrees.
VERSION	string	Version of the Meta-data (e.g., 3.0)
PROPID	string	Proposal ID (e.g., GBT/11A-043) that is generated from PROJID.
BAND	string	Observing band (e.g., X) that is generated from RECEIVER.
FILENAME	string	Filename in the format <i>YYYY_MM_DD_HH:MM:SS.fits</i>
FILESIZE	string	Computed via <code>os.path.getsize(filename)/1024.0</code>
MSGFLAG	string	Flags raised during meta-data curation due to aborted scans.
BADDATA	string	A flag to indicate that the data in question are bad (e.g., 1).

Table 4. AAT: Project

Column Name	Status	Data Type	GBT Mapping	Example
PROJECT_CODE	✓	varchar2	First part of PROJID	AGBT11B.010
OBSERVER	✓	varchar2(48)	OBSERVER	Karl Jansky
STARTTIME	✓	number	DATE-OBS to MJD	51862.01
TELESCOPE	✓	varchar2(12)	Constant value	GBT
OBS_BANDS	✓	varchar2(16)	From RECEIVER	L,X,C
NUM_SEGMENTS	✓	number	count directories	45 (for AGBT11B.010.*)
ARCH_FILES	✓	number	count all GO FITS files	112
RAW_PROJECT_CODE	✓	varchar2(16)	First part of PROJID	AGBT11B.010

Table 5. AAT: Archive

Column Name	Status	Data Type	GBT Mapping	Example
PROJECT_CODE	✓	varchar2	First part of PROJID	AGBT11B_010
SEGMENT	✓	varchar2(4)	Last part of PROJID	03
STARTTIME	✓	number	DATE-OBS to MJD	51862.01
TELESCOPE	✓	varchar2(12)	Constant value	GBT
ARCH_FORMAT	✓	varchar2(8)	File type	FITS
DATA_TYPE	✓	varchar2(12)	raw or calibrated	raw
ARCH_FILE	✓	varchar2(96)	file name	2006_12_05_22:25:30.fits
FILE_SIZE	✓	varchar2(96)	in kB	144589
RAW_PROJECT_CODE	✓	varchar2(16)	First part of PROJID	AGBT11B_010
OBS_BANDS	✓	varchar2(16)	From RECEIVER	L,X,C
PROCNAME	×	varchar2(16)	Procedure name	OffOn
PROCTYPE	×	varchar2(16)	Procedure Type	MAP

Table 6. AAT: Archfilelocation

Column Name	Status	Data Type	GBT Mapping	Example
SITE_CODE	✓	varchar2(8)	Location	CV
ROOT_DIRECTORY	✓	varchar2(128)	Archive disk array	/export/cvlustre/vulcan/
ARCH_FILE	✓	varchar2(96)	file name	2006_12_05_22:25:30.fits
MIMETYPE	✓	varchar2(80)	Data model	applicationx-gbt;datamodel=gbt-exp

Table 7. AAT: File_Set_Properties

Column Name	Status	Data Type	GBT Mapping	Example
FILE_SET_ID	✓	varchar2(128)	PROJID	AGB11B_010_03
FILE_SET_CATEGORY	✓	varchar2(32)	Observing type	Session
N_FILE.SUBSETS	✓	number	Files in obs session	23
COLLECTION	✓	varchar2(48)	Collection	GBT_ARCHIVE
CALIB_LEVEL	✓	number	(0) for raw	0
STARTTIME	✓	number	DATE-OBS to MJD	51862.01
TELESCOPE	✓	varchar2(12)	Constant value	GBT
OBS_BANDS	✓	varchar2(16)	RECEIVER	L,X,C

Table 8. AAT: Observation

Column Name	Status	Data Type	GBT Mapping	Example
PROJECT_CODE	✓	varchar2	First part of PROJID	AGBT11B.010
SEGMENT	✓	varchar2(4)	Last part of PROJID	03
OBS_TYPE	✓	varchar2(24)	OBSTYPE	LINE
STARTTIME	✓	number	DATE-OBS to MJD	51862.01
SOURCE_ID	✓	varchar2(16)	OBJECT	NGC4254
RA2000	✓	number	RA2000 (DB radians)	3.2393310
DEC2000	✓	number	DEC2000 (DB radians)	0.8482300
FRAME	✓	varchar2(20)	Sky Ref. Frame	ICRF
RA_EPOCH	✓	number	Epoch	2000.0
DEC_EPOCH	✓	number	Epoch	2000.0

Table 9. AAT: DataDesc

Column Name	Status	Data Type	GBT Mapping	Example
PROJECT_CODE	✓	varchar2(16)	First part of PROJID	AGBT11B.010
SEGMENT	✓	varchar2(4)	Last part of PROJID	03
RECEIVER	×	varchar2(16)	Receiver name	Rcvr8.10
BACKEND	×	varchar2(16)	Backend name	Spectrometer
IF_BAND	✓	number	From RECEIVER	L,X,C
IF_REF_FREQ	✓	number	RESTFn	1420405800.0
POL1	✓	number	Pol. code holder	1-20
POL2	✓	number	Pol. code holder	1-20
POL3	✓	number	Pol. code holder	1-20
POL4	✓	number	Pol. code holder	1-20
SUB_NUM_CHANS	✓	number	Chans in bandpass	4096
VELOCITY	×	number	Source velocity (m sec ⁻¹)	6542.23
VELDEF	×	varchar2(16)	Velocity definition	VRAD-LSR
VERSION	×	number	Meta-data version	3.0

Table 10. AAT: ObsSummary

Column Name	Status	Data Type	GBT Mapping	Example
PROJECT_CODE	✓	varchar2(16)	First part of PROJID	AGBT11B_010
SEGMENT	✓	varchar2(4)	Last part of PROJID	03
IF_BAND	✓	number	From RECEIVER	L,X,C
STARTTIME	✓	number	DATE-OBS to MJD	51862.01
TELESCOPE	✓	varchar2(12)	Constant value	GBT
SUB_NUM.CHANS	✓	number	Chans in bandpass	4096
RA2000	✓	number	RA2000 (DB radians)	3.2393310
DEC2000	✓	number	DEC2000 (DB radians)	0.8482300
RADEG	✓	number	RA2000 (DB degrees)	185.46
DECDEG	✓	number	DEC2000 (DB degrees)	48.6
OBSAZ	×	varchar2(16)	Azimuth (degrees)	172.56
OBSSEL	×	varchar2(16)	Elevation (degrees)	43.65

Table 11. Division of Labor

Topic	Personnel
§2 Data Storage	CIS (C. Clark, T. Halstead); OSO (D. Balser, B. Kent, G. Hunt)
§4 Meta-data	GB Software (TBD); OSO (D. Balser, B. Kent, G. Hunt)
§5 AAT	EVLA Computing (J. Benson); OSO (D. Balser, B. Kent, G. Hunt)
§7 Observing Policies	GB Software (TBD); OSO (D. Balser, B. Kent, G. Hunt)