A FITS Binary Table Convention for Interchange of Single Dish Data in Radio Astronomy

Harvey S. Liszt *

DRAFT of January 26, 1995

Abstract

This paper describes a FITS binary table convention for interchange of single dish data in radio astronomy, as derived from an agreement reached during an international meeting of interested parties at Green Bank, West Virginia in 1989 October. Following an introduction to the FITS image and binary table formats, we discuss the conventions necessary to implement the radio-astronomical single dish formalism.

1 Introduction

In 1989 October, representatives of many of the world's radio-astronomical observatories met in Green Bank West Virginia to consider the creation of a standard data transport and interchange format for so-called 'single dish' (i.e. non-synthesis) radioastronomy data. Also in attendance at the meeting were representatives of the community interested in and responsible for setting standards for the FITS data format(s), as well as a few X-ray any other astronomers with a recognized need to exchange data not well described as standard FITS 'images'.

Seeking a universal and efficient means of writing their data, those assembled were fortunate in being able to capitalize on the emerging FITS binary table format presented by W. Cotton of the NRAO. They enthusiastically endorsed the use of FITS '3-D' tables, as they were known at the time, following their internal use within the AIPS package. The feasibility of such an implementation was demonstrated by the prototype data readers and writers constructed by F. Ghigo (NRAO) and T. Forveille (IRAM and U. Grenoble). A version of the standard was implemented at the NRAO as its standard export format in early 1992. This paper describes the fruits of the labors of these workers, in the hope that it will be widely adopted by the community which met to create it.

This work follows the tradition set earlier by those papers which defined the FITS image (Wells, Greisen, and Harten 1981, [5]) and other format (c.f. Greisen and Harten 1981, [2], for random groups; Grosböl, *et al.*1988, [3], for extensions and Harten *et al.*1988, [4], for the original FITS tables). However, this paper is also a departure in that it only describes the agreed-upon *use* of an existing, albeit very new, FITS format: it does not define the binary-table format itself, which is described in (Cottoon, Tody, and Pence 1995 [1]). A general familiarity with FITS usage is needed to understand this paper. However, the hope is to provide enough information to implement a version of the single dish FITS (SDF) convention even in the absence of a full description of the underlying FITS binary tables.

2 Binary Tables *vs.* images

The success of the original FITS image format was due to its ability to describe large quantities of regular-gridded data in a clear and succinct fashion. The structure of FITS images, sketched schematically in the case of a two-dimensional image in figure 1, lend themselves to both human and machine translation with

*National Radio Astronomy Observatory, Charlottesville, VA 22903-2475

```
NAXIS   =                    2
BITPIX  =
NAXIS1  =
.
.
.
NAXIS2  =
.
.
.
END
```

```
     1st              2nd              3rd           NAXIS2-th
NAXIS1 pixels    NAXIS1 pixels    NAXIS1 pixels ... NAXIS1 pixels
```

Figure 1: Schematic view of a FITS Image. Within the header, a set of 36x80-byte ASCII card-images, the number of points along each of NAXIS axes are specified by the NAXISnnn keywords. Within the data blocks, which follow the header, the sequential binary-formatted data points are implicitly blocked into a FORTRAN array (naxis1, naxis2). Each data point is in a standard format encoded in the BITPIX value. The data occupy $1 + (\mathtt{NAXIS1}\,\mathtt{NAXIS2}\,(|\mathtt{BITPIX}|\,/8) - 1)/2880)$ 2880-byte data blocks.

a minimum of overhead because of the regularity of the data and the few additional parameters needed to unpack the image and its description (typically some measure of source strength *vs.* sky coordinates). Even though the KEYWORD=VALUE scheme is relatively inefficient on a line-by-line basis in the header section of an 80-byte card image blocked into blank-padded 2880-byte blocks, the volume of image data is typically so large that the wasted space is inconsequential.

Clearly the image format is inefficient when the volume of data is small compared to the amount of header information which is present (for whatever reason). For this reason, an early variant of the FITS standard known as random groups was implemented to hold *uv*-data for synthesis observations (Greisen and Harten 1981). However, this mechanism did not capture the imagination of the community and its use is now deprecated, although the basic concept is widely employed elsewhere (as in .WK? spreadsheet files). The FITS binary table extension was designed to exploit greater use of binary number representations rather than the textual KEYWORD=VALUE scheme. It avoids using a full 80 bytes for each KEYWORD=VALUE instance and exploits the regularities present when even very complex or irregular data structures are written *repetitively.*

The FITS binary table extension is a means of writing complex data structures directly to external media using (nearly) arbitrary aggregations of standard data types (ints, longints, IEEE real or double, char[] etc.). In the example schematized in figure 2, a series of header variables, each filling part of one column of the table, precedes a data matrix. By writing the same structure over and over with different values, the table acquires its so-called two-dimensional symbolic form. However, the entities making up the symbolic table can themselves be dimensioned as long as they are composed of identical atomic members. This is obvious for the case where a column is a character string, *i.e.* an array of characters, if less so for a vector of numbers.

In either case, one refers to the column as its constituents ("that column is the data") or by their makeup ("that column is an array of long integers"). The intersection of a row and column, an instance of one of the constituents of the table, is referred to as a field or being contained in a field. It is clear that extension of the table concept to contain whole tables in individual fields is still consistent with the symbolic two-dimensional cell form sketched in figure 2. While defined for binary tables in general, this is beyond the scope of the present discussion.

Before proceeding, it is important to note that the FITS binary table convention applies to the somewhat limited case where all rows of the table have the same structure and length.

```
            1            2            3                TFIELDS
1           header var   header var   header var   ... data matrix
2           header var   header var   header var   ... data matrix
.           .            .            .                .
.           .            .            .                .
.           .            .            .                .
NAXIS2      header var   header var   header var   ... data matrix
```

Figure 2: A schematic view of a FITS binary table body. The table is implicitly blocked into NAXIS2 instances (rows) of TFIELDS quantities written in binary form (the precise use of this jargon is explained in the text). The locations, description, and binary format for each field must be given in the header of the table, requiring an extra layer of indirection. All rows have identical structures, but different contents. The header variables do not all have the same byte-count.

```
SIMPLE  =                    T  / File conforms to FITS standards
BITPIX  =                    8
NAXIS   =                    0  / No image data array present
EXTEND  =                    T  / Standard extensions follow
BLOCKED =                    T  / Tape block may be 1-10 times 2880 bytes
ORIGIN  = 'UNIPOPS u2f/1.4'     / Written by UniPOPS u2f utility
DATE    = '12/12/93'            / Date (dd/mm/yy) this file was written
COMMENT    Single dish data from Green Bank
END
```

Figure 3: First header block of a FITS file. There is no data immediately following (NAXIS=0), but an extension (EXTEND=T) is present (see figure 4).

3 FITS image and binary table header structures

In standard FITS images, a header composed of KEYWORD=VALUE pairs has two responsibilities. Its direct function is to convey the data structure (flux *vs.* right ascension and declination, for instance). This is accomplished by giving the number of axes in the data with the mandatory NAXIS and NAXISnn keywords, together with the regular coordinate information along each axis (using the CRVALnnn, CRPIXnnn and CTYPEnnn keywords). The data are then understood to be unpacked following the FORTRAN convention that the first axis has the most rapidly-varying index. But the header also must describe the format in which the data values are presented, with the BITPIX keyword. Standard values of this imply scaled integers (BITPIX=16), floats (BITPIX=-32), etc.

There is essentially one layer of indirection in the FITS image standard. Following the line SIMPLE=T (which guarantees a FITS file of some kind) is a set of header keywords in column one and the FITS image reader parses the input lines until an END statement is encountered, at which point the data matrix begins at the next 2880 byte block boundary. The structure of the data in both machine form and on the sky is known and read. FITS binary tables require more layers of indirection due to the irregular nature of the fields preceding the data matrix.

A typical FITS header announcing that extension(s) follow is shown in (figure 3). This header is likely to be typical for most SDF data in that there is no primary image (NAXIS=0). The SDF convention applies only to the contents of the FITS binary table. This convention does not rule out the possibilty that some SDF data may have an associated primary image (*e.g.*,an image produced from the associated SDF table). A single FITS file may contain multiple tables, any number of which may obey the SDF convention.

A subsequent header block (figure 4) announces the arrival of a binary table in the lines below. The presence and ordering of the first nine KEYWORD=VALUE pairs are obligatory, the first eight due to the binary table standard and the ninth to the SINGLE DISH convention. The BITPIX=8, NAXIS=2,

```
XTENSION= 'BINTABLE'           / FITS binary table
BITPIX  =                    8 / MANDATORY--Binary data.
NAXIS   =                    2 / MANDATORY--A 2D TABLE!
NAXIS1  =                 3320 / MANDATORY--width of table in bytes
NAXIS2  =                   18 / MANDATORY--Number of rows in table
PCOUNT  =                    0 / MANDATORY, Recommended value 0,no heap
GCOUNT  =                    1 / MANDATORY
TFIELDS =                   13 / MANDATORY--Number of fields per row
EXTNAME = 'SINGLE DISH'        / MANDATORY--Single dish convention
EXTVER  =                    3 / ENTIRELY OPTIONAL
EXTLEVEL=                    1 / OPTIONAL, SHOULD DEFAULT TO 1
NMATRIX =                    1 / PRESENCE MANDATORY, 1 DATASET PER ROW
```

Figure 4: Start of the header block of a FITS binary table following the SDF convention. Much of the structure shown here is mandatory (see the text).

```
NAXIS   =                    5
NAXIS1  =                  256
CTYPE1  = 'FELO-LSR'
CRPIX1  =                  128
CRVAL1  =      -190.0000000E+05
CDELT1  =         5.2000000E+05
NAXIS2  =                    1
CTYPE2  = 'RA---SIN'
CRPIX2  =                    1
CRVAL2  =       -28.99900000000
CDELT2  =                    0
```

Figure 5: FITS image specification of the axes of a data matrix using the NAXIS, NAXISnnn keywords.

and GCOUNT=1 are mandatory *assignments*, as well as is EXTNAME='SINGLE DISH' describing the convention. The PCOUNT=0 is recommended for binary tables and NAXIS1=3320 informs the reader that (in this case) each row of the column has a total length of 3320 bytes comprising TFIELDS=13 fields (one of these fields will be the actual data matrix). The EXTVER and EXTLEVEL are optional descriptions. The last keyword in this section of the header states that there will (in this case) be a single data matrix. The presence of this keyword is mandatory, and the value must be at least 1. This is the only keyword peculiar to the SDF convention in this prelude.

The remaining part of the header (figure 6) is where the extra level of indirection arises as cited above; because the entries in the table will be otherwise-anonymous binary numbers, the content and form of the fields of the table must be specified for later unpacking of the actual values. Consider first the specification of the data matrix. Within the context of the FITS image format, the number of axes, NAXIS, and the number of pixels along the first two axes (NAXIS1 and NAXIS2)could be specified by lines as show in figure 5.

These lines might pertain to a 256-channel spectrum whose frequency-gridded velocity axis has the fiducial value -190 km s^{-1} in the center of channel 128, taken at the stated right ascension with declination and line rest frequency filling out the remaining degenerate or one-pixel axes (not shown here). These same fields will appear serially in binary form in the table with a mixture of ASCII, integer, and floating formats without description, so one must be provided here. We do this in the following way.

4 Encapsulating a data matrix within binary tables

First, because the NAXIS and NAXISnnn keywords of FITS images have been preempted by the table header, they are replaced by SDF-specific MAXIS and MAXISnnn keywords having the same meaning (or by another mechanism discussed later). Then we use the standard binary table TFORMnnn, TTYPEnnn, and TUNITnnn keywords to put in the header the equivalent of the the *left-hand sides* of the KEYWORD=VALUE pairs cited just before. The actual values, the right-hand sides, will come later, perhaps many times, in the body of the table.

4.1 Repeat counts and standard data types

The contents of the table fields are specified in lines like

```
TFORMnnn='rptQ'
```

where rpt is an integer repeat count describing the number of repetitions of the atomic data type Q. Recognized values of Q include A (char), L (logical), X (bit), B (unsigned byte), I (signed 16-bit two's compliment integer), J (32-bit integer), E (4-byte IEEE real), D (8-byte real), C (single precision complex) and M (double precision complex).

4.2 An example data matrix

To embed the formal description of the data matrix within the table header, we use lines like those shown in figure 6. These lines inform the reader that the data matrix has five axes (four will be degenerate). Then the first field in each row will be an array of eight characters (having the value 'FELO-LSR') which are to be interpreted as the usual CTYPE1. This will be followed by a single integer to be interpreted as MAXIS1 (the value will be 256 later). The third field in the row will be a double-precision (IEEE 8-byte) real representing the CRVAL1 required to specify the fiducial velocity, etc. This part of the header will continue until all MAXIS=5 axes have been specified in sufficient detail that the reader will b e able to function when it actually encounters them in the table.

The actual MAXIS-dimensional data matrix will occupy one true column somewhere in the table. It also requires descriptors, which are illustrated in figure 7.

The TTYPEnnn='DATA' for the field containing the data is mandatory, as is the TMATXnnn=T. The value TMATXnnn=F is understood for all axes that are not arrays of values; the TTYPEnnn axis announces the arrival of the actual data array (there could be arrays which are not the data). Character strings are a special case of arrays which are specified only in the TFORMnnn parameter.

In all, these lines enable the reader to understand a chunk of binary-formatted data which is actually blocked as in figure 8.

4.3 Number and kind of columns

Implicit in the immediately-preceding discussion is a built-in efficiency based on the global quality of certain values. Because the MAXIS and MAXISnnn values are the same for all table columns, they may be abstracted and written once in the table header rather than repeated. Thus these (and other) global parameters are written once and become virtual rather than true columns in the table. In the example above, the CTYPE1 and MAXIS1 keywords are present in the table as true columns but at least the second of these must by convention never change its value. It might be of some use to intersperse different kinds of arrays with the same dimensions and byte count, as could be the case in the preceding example.

Other examples of global parameters which will often be virtual are shown in figure 9 (this convention does not require these parameters to be global).

In general, every header word embedded in the body of the binary table requires a TTYPEnnn=, TFORMnnn= pair in the header, (TUNITnnn is optional). Header words which are global are expressed directly. Header words which are absent, so-called non-existent columns, will have to be given reasonable defaults by the reader.

```
MAXIS   =                    5
TFORM1  = '8A       '
TTYPE1  = 'CTYPE1   '
TFORM2  = '1I       '
TTYPE2  = 'MAXIS1   '
TFORM3  = '1D       '
TTYPE3  = 'CRVAL1   '
TUNIT3  = 'M/SEC    '
TFORM4  = '1E       '
TTYPE4  = 'CRPIX1   '
TFORM5  = '1E       '
TTYPE5  = 'CDELT1   '
TUNIT5  = 'M/SEC    '
MAXIS2  =                    1
TFORM6  = '8A       '
TTYPE6  = 'CTYPE2   '
TFORM7  = '1D       '
TUNIT7  = 'DEGREES  '
TFORM8  = '1E       '
TTYPE8  = 'CRPIX2   '
TFORM9  = '1E       '
TTYPE9  = 'CDELT2   '
TUNIT9  = 'DEGREES  '
[specs for 3 other header words]
```

Figure 6: Specification of the axes of the data matrix within the second header block of a FITS binary table.

```
TTYPE13 = 'DATA     '
TFORM13 = '256E     '
TUNIT13 = 'KELVIN   '
TMATX13 =                    T
```

Figure 7: Specification of the location of the data matrix in the 13th table field (TMATX13 = T), and the format of the data contents (256E in this example, the type of the data column could be any of the allowed binary table types).

```
            <-----------------------NAXIS1=3320 bytes----------------->
                 1   2    3     4  5       6                 TFIELDS=13
1st row     FELO-LSR256-190E+071285.2E05RA---SIN...................256E
2nd row
.
.
.
NAXIS2=18th FELO-LSR256-0.1E+051630.2E05RA---SIN...................256E
```

Figure 8: A schematic ASCII representation of a binary table showing the meaning of NAXIS1 and NAXIS2. The type of the data column need not be E, IEEE floating point.

```
TELESCOP= 'NRAO 43M'
OBSERVER= 'YOUR AD GOES HERE'
DATE-OBS= '09/12/93'
DATAMAX =            4.0019E+01
DATAMIN =           -1.8797E+02
```

Figure 9: Other keywords which will often be global.

4.4 Alternate specification of data matrix dimensions.

Since the MAXIS keyword convention was adopted at the Green Bank meeting, a more compact notation has come into use within the context of FITS binary tables. The MAXIS and MAXISnnn specifications could be replaced by the single line

```
TDIMnnn=(MAXIS1,MAXIS2,...)
```

where actual numbers corresponding as shown are included; TDIM13=(256,1,1,1,1) would be appropriate for the above example with MAXIS=5. The TDIMnnn keyword would probably be grouped with the rest of the TXXXnnn keywords for the data matrix. Use of TDIMnnn rather than MAXIS might render SDF tables more accessible to the extant body of table readers.

5 Keywords and description of the data and its origins

Much of the time at the Green Bank meeting was devoted to identifying a set of common and ubiquitous keywords. At that time, it seemed likely that FITS would adopt a set of hierarchical name-spaces supporting discipline-oriented dictionaries of targeted keywords. Alas, this proved too onerous and was abandoned. The immediate consequence of this inaction is simply that the KEYWORD=VALUE scheme continues as before and no new semantics need be considered to read or write SDF tables. Those portions of the draft summary of the 1989 SDF meeting dealing with hierarchical keywords can simply be abandoned.

Whenever possible, keywords should be chosen from a pre-existing set instead of inventing new ways to specify any quantity whose meaning is well-defined in basic FITS, but in general, keywords for the SDF convention do not exist in widely-accepted form.

One set of rules for selecting keyword descriptors might be; a) retain whole words or concatenate two words in their entirety if they have $4 \leq n \leq 8$ characters, i.e., pressure represented by "PRESSURE", Dew Point by "DEWPOINT"; b) truncate all single words to 8 characters if necessary, i.e., bandwidth represented by "BANDWIDT"; c) concatenate two words and truncate the result if the first word is sufficiently short that the second word is not obliterated, i.e. cycle length represented by "CYCLELEN"; d) concatenate double words selecting several letters from each if a,b,c fail, i.e., aperture efficiency represented by "APEREFFI", e) improvise, making sure to stay within reasonable upper and lower bounds on the length and intelligibility of the keyword. Whichever rule is used, the limit of 8 characters must be followed since keywords can either be global (where the FITS rules require that keywords have 8 or fewer characters) or column names (where the FITS rules would allow longer names).

It was agreed that parameters of the observations fall into three classes.

5.1 Core keywords and description of the data origins and axes

The core keywords were held to be essential and common to all observations and telescopes. These are described by a set of universal keywords; making up a different keyword or meaning for a core parameter is "illegal". All SDF readers/writers *must* write and properly interpret *all* core keywords. The core keywords are not intended to be comprehensive, in the sense that other quantities required to describe the observations(axis descriptors, etc.) are not included. However, to the extent that subtleties of the data may not be otherwise conveyed (i.e. the channel spacing may differ from the spectral resolution), the core keywords provide

```
OBJECT   = '3C454.1 '               / common FITS usage
TELESCOP= 'BONN-100M'               / common FITS keyword
FREQRES =        1.44923239E03      / resolution may differ from spacing
BANDWID =        1.47323223E06      / of backend, not one channel
DATE-OBS= 'DD/MM/YY'                / common FITS usage; JD preferable?
TIME    =        1.34230000E03      / UT time of day; UT seconds since 0h UT
EXPOSURE=            1.4400E05      / effective integration time
TSYS    =        2.43120000E02      / system, not receiver, temperature
```

Figure 10: The set of "CORE" keywords required of any SDF implementation in addition to the standard axis information keywords described earlier (see Figure 6). For convenience, they are shown here when appearing as virtual columns in the header of the table. In most cases, they would instead appear as real columns in the table body. Note that TIME coule also be a formal axis of the data (see the full example in the appendix).

additional necessary information. Membership in the CORE class of keywords was specified to include keywords for the quantities illustrated in figure 10. For convenience, they are shown in the figure as appearing as virtual columns in the header of the table; in most cases, they would instead appear as real columns in the table body.

Within the FITS framework, description of sky coordinates is now highly refined following the work of Greisen and Calabretta (cite here). Specification of the velocity or frequency axis required of spectra is a thornier problem which requires clarification. The frequency of velocity interval between spectrometer channels will be described by the basic FITS CRDELT value; although it is possible for the channel frequency spacing to be a function of channel ordinate, it was at least tentatively agreed that modern AOS devices should be regridded onto a linear scale before presentation to the observer. But the data may be over- or under-sampled such that a new core keyword (FREQRES) was introduced to describe the width of spectral channels as distinct from their spacing. In turn, the width (FREQRES) and spacing (CRDELT) are different from the backend bandwidth (BANDWIDT).

To the extent that spectrometers are gridded in frequency, all velocity descriptions of such data are inherently non-linear, whence the "FELO" FITS description. The harm done in adhering to a linear description in velocity may be small, in which case the rest frequency of the emissions will typically be appended to the data as a degenerate axis. The shared keywords provide the VELDEF keyword to specify the definition and reference frame of the velocity.

The system temperature, TSYS, effective integration time (EXPOSURE), and frequency resolution included among the core keywords are to be interpreted such that the radiometer equation

$$\mathtt{TSYS}\,/\sqrt{\mathtt{EXPOSURE} \times \mathtt{FREQRES}}$$

is directly proportional to the rms noise. If a receiver temperature is given using the shared keyword TRX, it and TSYS should agree in the sense of being single or double sideband values.

5.2 Shared keywords selected at the Green Bank meeting.

During the meeting, those assembled selected a list of important parameters which will be described by the shared class of keywords. These were held to be largely common to all observations and telescopes but not essential. They will have one legal descriptor and meaning – but these may be ignored by a SDF reader/writer. The presence of shared keywords is not mandatory but inventing a descriptor or meaning for any of them is "illegal".

A very few of these are well-defined FITS keywords, many had direct analogs in the older general category of a now-extinct NRAO ASCII-FITS table format, and others (image frequency, atmospheric opacity) required new definitions entirely. All quantities with temperature units are quoted in Kelvin and all quantities which are fractions or percentages are now defined as reals in the range 0..1. In figure 11 the contents of the shared class defined at the 1989 meeting are shown as for those in the core group. Most of the entries are obvious. Some have options which are discussed in the figure caption.

5.3 Site-specific keywords.

Many observational parameters may be unique to a given site and beyond specification by a committee. Keywords for such parameters may be invented at will as long as they do not duplicate any already in use as core or shared keywords. Assignments should be made using standard units (degrees of arc, Kelvin, fractions expressed in the range [0..1], etc.) SDF readers must be prepared to disregard these if they cannot be used, and their presence is entirely optional.

Acknowledgments

The National Radio Astronomy Observatory (NRAO) is operated by Associated Universities Inc, under cooperative agreement with the National Science Foundation.

```
OBSERVER= 'HOGG    '             / Observer name; A-format
OBSID   = 'DEH WS  '             / Observer & operator initials; A-format
PROJID  = 'TEST    '             / Project ID; A-format
SCAN    =       4.87800000E+03  / Scan number; Float
OBSMODE = 'CONTBMSW'             / Type of data, observing mode;8A
MOLECULE= 'methyl fornicate'     / Helpful description; A-format
TRANSITI= 'v=0 j=9-8'            / As appropriate; A-format
TEMPSCAL= 'TA*     '             / Normalization of TA;A-format
FRONTEND= 'SiS 3mm '             / A-format
TCAL    =            8.0000000  / Calibration Temp (K); Float
THOT    =        2.4400000E+02  / Hot load temperature (K); Float
TCOLD   =        7.1500000E+00  / Cold load temperature (K); Float
TRX     =               340.00  / Receiver Temp (K), Float
VELDEF  = 'RADI-LSR'             / Velocity definition & frame; A-format
VCORR   =       0.10020000E+00  / radial velocity correction; Vref - Vtel
OBSFREQ =       0.86243350E+11  / Observed Frequency (Hz); Float
IMAGFREQ=       0.92000000E+09  / Image sideband freq (Hz); Float
LST     =       2.36953330E+00  / LST (seconds) at start of scan; Float
LST     =       8.53031880E+03  / LST (seconds) at start of scan; Float
AZIMUTH =       2.34865730E+00  / Commanded Azimuth (Degrees); Float
ELEVATIO=       1.85010360E+00  / Commanded Elevation(Degrees); Float
TAU     =       0.24297244E+00  / Opacity at signal freq; Float
TAUIMAGE=       0.24297244E+00  / Opacity at image freq; Float
TAUZENIT=       0.14274444E+00  / Opacity per unit air mass; Float
HUMIDITY=       0.16953330E+00  / Decimal fraction 0..1; Float
TAMBIENT=       2.92000000E+02  / Ambient Temp (K); Float
PRESSURE=       0.97100000E+04  / Barometer reading mm Hg; Float
DEWPOINT=       2.73000000E+02  / Dew point (K); Float
WINDSPEE=       2.30000000E+00  / Wind speed m/s; Float
WINDDIRE=       2.30000000E+00  / Degrees West of North; Float
BEAMEFF =       0.91600000E+00  / Main-beam efficiency; Float
APEREFF =       0.44999999E+00  / Antenna Aperature Efficiency; Float
ETAL    =       0.92430000E-01  / Rear spillover; Float
ETAFSS  =       0.87230000E+00  / Accounts for forward loss; Float
ANTGAIN =       0.13400000E+00  / K per Jy; Float
BMAJ    =       3.50000000E+01  / Large main-beam FWHM; Float
BMIN    =       3.50000000E+01  / Small main-beam FWHM; Float
BPA     =       2.30000000E+01  / Beam position angle; Float
SITELONG=       7.84523000E+01  / Site longitude (Degrees); Float
SITELAT =       1.82452000E+01  / Site latitude (Degrees); Float
SITEELEV=       6.02340000E+02  / site elevation in meters; Float
EPOCH   =       1.95000000E+03  / Epoch of observation (year); Float
EQUINOX =       1.95000000E+03  / Equinox of coords (year); Float
```

Figure 11: The SHARED keyword class. `OBSMODE` : Type of data–LINE, CONT, PULS (*et al.*) + Mode–PSSW (position switch), FQSW (frequency switch), BMSW (beam switch), PLSW (phase-lock switch), LDSW (load switch), TLPW (total power); `TEMPSCAL` TB,TA,TA*TR,TR*,etc.; `TRX` should be SSB, DSB as for TSYS; `VELDEF` : 'RADI','OPTI','RELA'+'LSR','HELO','EART',BARI','-OBS'; `BPA` is measured East of North

A An example SDF header.

This appendix contains an example of a complete FITS binary table header conforming to the SDF convention. The TDIMnnn convention is used rather than the MAXIS keyword convention. Note that some of the axis description information (CRVALn, CRPIXn, and CDELTn) is virtual (a single value applying to the entire table) while most is contained in columns (each value in a field only applying to a specific row). There are no site-specific keywords shown in this example. This convention does not require any specific ordering of the keywords nor does it specify the format (value of TFORMnnn keyword) for any columns except for A format columns. All of the SHARED keywords are *not* used in this example.

```
XTENSION= 'BINTABLE'                    / FITS binary table.
BITPIX  =                     8         / Binary data.
NAXIS   =                     2         / Table is a Matrix.
NAXIS1  =                  2332         / Width of table in bytes.
NAXIS2  =                    48         / Number of entries (rows) in Table.
PCOUNT  =                     0         / No Random Parameters.
GCOUNT  =                     1         / Only one group.
TFIELDS =                    49         / Number of fields in each row.
EXTNAME = 'SINGLE DISH'                 / Single Dish Convention.
EXTVER  =                     1         / Version number of Table.
NMATRIX =                     1         / No.matrixes following SD convention.
           / The following global keywords apply, at least, to the first scan.
TELESCOP= 'NRAO 43M          '          / Designation of Telescope.
OBSERVER= 'GARWOOD           '          / Name of observer.
DATE-OBS= '14/08/92          '          / UT date of observation (dd/mm/yy).
DATAMAX =            1.2773E+02         / Max spectral value (K) - for whole file.
DATAMIN =           -9.3381E+00         / Min spectral value (K) - for whole file.
           / The following defines the data matrix.
CTYPE1  = 'FREQ    '                    / Frequency Axis.
TFORM1  = '1D      '                    /
TTYPE1  = 'CRVAL1  '                    / Frequency at reference Pixel.
TUNIT1  = 'HZ      '                    /
TFORM2  = '1E      '                    /
TTYPE2  = 'CRPIX1  '                    / Frequency Reference Pixel.
TFORM3  = '1E      '                    /
TTYPE3  = 'CDELT1  '                    / Frequency increment.
TUNIT3  = 'HZ      '                    /
TFORM4  = '8A      '                    /
TTYPE4  = 'CTYPE2  '                    / Name of longitude-like axis.
TFORM4  = '1D      '                    /
TTYPE5  = 'CRVAL2  '                    / Value at longitude reference pixel.
TUNIT5  = 'DEGREES '                    /
TFORM6  = '1E      '                    /
TTYPE6  = 'CRPIX2  '                    / Longitude reference pixel.
TFORM7  = '1E      '                    /
TTYPE7  = 'CDELT2  '                    / Longitude increment.
TUNIT7  = 'DEGREES '                    /
TFORM8  = '8A      '                    /
TTYPE8  = 'CTYPE3  '                    / Name of latitude-like axis.
TFORM9  = '1D      '                    /
TTYPE9  = 'CRVAL3  '                    / Value at latitude reference pixel.
TUNIT9  = 'DEGREES '                    /
TFORM10 = '1E      '                    /
```

```
TTYPE10 = 'CRPIX3  '                 / Latitude reference pixel.
TFORM11 = '1E      '                 /
TTYPE11 = 'CDELT3  '                 / Latitude increment.
TUNIT11 = 'DEGREES '                 /
TFORM12 = '1E      '                 /
TTYPE12 = 'CROTA3  '                 / Rotation of Axes.
TUNIT12 = 'DEGREES '                 /
CTYPE4  = 'TIME    '                 /
CRPIX4  =                  1.0       / Reference pixel on time axis.
TFORM13 = '1E      '                 /
TTYPE13 = 'CRVAL4  '                 / UT time in seconds.
TUNIT13 = 'SECONDS '                 /
CTYPE5  = 'RECEIVER'                 /
CRPIX5  =                  1.0       / Reference pixel on RECEIVER axis.
TFORM14 = '1E      '                 /
TTYPE14 = 'CRVAL5  '                 / RECEIVER or BEAM number.
            /The remaining Single Dish CORE keywords follow.
TFORM15 = '16A     '                 /
TTYPE15 = 'OBJECT  '                 / Observed object name.
TFORM16 = '8A      '                 /
TTYPE16 = 'TELESCOP'                 / Telescope designation.
TFORM17 = '1E        '               /
TTYPE17 = 'FREQRES      '            / Frequency resolution of a channel.
TUNIT17 = 'HZ           '            /
TFORM18 = '1E        '               /
TTYPE18 = 'BANDWIDT     '            / Total Receiver Bandwidth.
TUNIT18 = 'HZ           '            /
TFORM19 = '8A      '                 /
TTYPE19 = 'DATE-OBS'                 / Date of observation in form: yy/mm/dd
TFORM20 = '1E      '                 /
TTYPE20 = 'EXPOSURE'                 / Effective on-source integration time.
TUNIT20 = 'SECONDS '                 /
TFORM21 = '1E      '                 /
TTYPE21 = 'TSYS    '                 / System Temperature on source.
TUNIT21 = 'KELVIN  '                 /
            / The following are single-dish SHARED keywords.
TFORM22 = '16A     '                 /
TTYPE22 = 'OBSERVER'                 / Observer's name
TFORM23 = '8A      '                 /
TTYPE23 = 'OBSID   '                 / Observer's ID code.
TFORM24 = '8A      '                 /
TTYPE24 = 'PROJID  '                 / Project ID code
TFORM25 = '1D      '                 /
TTYPE25 = 'SCAN    '                 / Scan number (frac.part may be rcvr no.)
TFORM26 = '8A      '                 /
TTYPE26 = 'OBSMODE '                 / Observing mode.
TFORM27 = '8A      '                 /
TTYPE27 = 'FRONTEND'                 / Front-end receiver description.
TFORM28 = '1E      '                 /
TTYPE28 = 'TCAL    '                 / Calibration Temperature.
TUNIT28 = 'KELVIN  '                 /
TFORM29 = '1E      '                 /
TTYPE29 = 'TRX     '                 / Receiver Temperature.
```

```
TUNIT29 = 'KELVIN  '                    /
TFORM30 = '8A      '                    /
TTYPE30 = 'VELDEF  '                    / Velocity Definition and reference.
TFORM31 = '1E      '                    /
TTYPE31 = 'RVSYS   '                    / Velocity Correction, Vref-Vtel.
TUNIT31 = 'M/SEC   '                    /
TFORM32 = '1D      '                    /
TTYPE32 = 'RESTFREQ'                    / Rest Frequency at ref. pixel.
TUNIT32 = 'HZ      '                    /
TFORM33 = '1D      '                    /
TTYPE33 = 'OBSFREQ '                    / Sky Frequency at ref. pixel.
TUNIT33 = 'HZ      '                    /
TFORM34 = '1E      '                    /
TTYPE34 = 'LST     '                    / Local sidereal time.
TUNIT34 = 'SECONDS '                    /
TFORM35 = '1D      '                    /
TTYPE35 = 'AZIMUTH '                    / Commanded azimuth.
TUNIT35 = 'DEGREES '                    /
TFORM36 = '1D      '                    /
TTYPE36 = 'ELEVATIO'                    / Commanded elevation angle.
TUNIT36 = 'DEGREES '                    /
TFORM37 = '1E      '                    /
TTYPE37 = 'HUMIDITY'                    / Relative Humidity (fraction).
TFORM38 = '1E      '                    /
TTYPE38 = 'TAMBIENT'                    / Ambient Temperature.
TUNIT38 = 'KELVIN  '                    /
TFORM39 = '1E      '                    /
TTYPE39 = 'PRESSURE'                    / Atmospheric Pressure in Pascals.
TUNIT39 = 'PASCAL  '                    /
TFORM40 = '1E      '                    /
TTYPE40 = 'DEWPOINT'                    / DewPoint (Kelvin).
TUNIT40 = 'KELVIN  '                    /
TFORM41 = '1E      '                    /
TTYPE41 = 'BEAMEFF '                    / Main Beam Efficiency.
TFORM42 = '1E      '                    /
TTYPE42 = 'APEREFF '                    / Aperture Efficiency.
TFORM43 = '1E      '                    /
TTYPE43 = 'ETAL    '                    / Rear spillover and scattering eff.
TFORM44 = '1E      '                    /
TTYPE44 = 'ETAFSS  '                    / Forward spillover and scattering eff.
TFORM45 = '1E      '                    /
TTYPE45 = 'ANTGAIN '                    / Antenna Gain, Kelvins per Jy.
TUNIT45 = 'K/JY    '                    /
TFORM46 = '1E      '                    /
TTYPE46 = 'EQUINOX '                    / Equinox of position ref. frame.
TUNIT46 = 'YEAR    '                    /
            / Not required, but useful information
TFORM47 = '1E      '                    /
TTYPE47 = 'DATAMAX '                    / Max value in series.
TFORM48 = '1E      '                    /
TTYPE48 = 'DATAMIN '                    / Min value in series.
            / Here is the data matrix.
TFORM49 = '512E    '                    /
```

```
TTYPE49 = 'DATA    '                / Observed Data
TUNIT49 = 'KELVIN  '                /
TMATX49 =                    T      / This is a matrix.
END
```

References

[1] Cotton, W.D., Tody, D., and Pence, W.D. 1995, "Binary Table Extension to FITS" preprint..

[2] Greisen, E.W. and Harten, R.H. 1981, "An Extension of FITS for Groups of Small Arrays of Data," *Astr. &Ap. Suppl. Ser.*, **44**, 371.

[3] Grosbö l, P., Harten, R.H., Greisen, E.W., and Wells, D.C. 1988, *A&A Suppl.*, 73, 359.

[4] Harten, R.H., Grosböl, P., Greisen, E.W., and Wells, D.C. 1988, *A&A Suppl.*, 73, 365.

[5] Wells, D.C., Greisen, E.W., and Harten, R.H. 1981, *A&A Suppl.*, 44 363.