Robert C. Byrd Green Bank Telescope
NRAO Green Bank

Garwood, Roshi, Brandt

December 11, 2017

GBT SOFTWARE PROJECT NOTE 33.2

GBT VEGAS FITS File Specification

HTML version Available[1]

Contents

Abstract

The FITS format structure is presented for the GBT VEGAS scan data files. The scan data FITS files are permanently archived after each observation, and will usually be input to a GBT filler, which has the task of combining all scan data FITS files into some other form (e.g. SDFITS). This data may then be processed further in an analysis package (e.g. GBTIDL).

The scan data FITS files contain both 1) a complete description of the VEGAS setup for the observation and 2) measurement data with a FITS keyword description of the data organization. The `ACT_STATE`, `PORT`, `STATE` and `SAMPLER` tables describe the setup and the `DATA` table contains measured data values (spectra). The `SPURS`table contains a list of known channels affected by ADC spurs.

[1] http://www.gb.nrao.edu/GBT/MC/doc/dataproc/gbtVEGASFits/gbtVEGASFits/gbtVEGASFits.html

History

11 July 2011 Initial draft (Bob Garwood).

21 July 2011 Draft ready for wider comment (Bob Garwood).

12 April 2012 Verion 1.0 is set by this document. VEGAS FITS files produced on this date have been verified to agree with this document (Bob Garwood).

23 August 2013 Updated to include changes during initial commissioning. Future changes will be accompanied by a change to FITSVER. In the PHDU, BOFFILE keyword noted, presence of VEGAS status keywords noted. In the DATA table, the un-normalized state of the DATA column was noted and 5 additional columns were documented. A more recent VEGAS FITS file was used as the source of the examples shown here. (Bob Garwood)

11 December 2017 Version 1.2. Added a table of known channels where ADC spurs are predicted for each IF listed in the SAMPLER table. Added descriptions of additional PRIMARY header keywords. Revised the note about normalization of data values.

1 Background

All GBT scan data FITS files should conform to the standards specified in the GBT Software Project Note 4.0, "Device and Log FITS Files for the GBT." The VErsatile GBT Astronomical Spectrometer (VEGAS) consists of 8 spectrometers. Each spectrometer is called a `BANK` and the data from each `BANK` is writen to a separate FITS file. This document describes the structure of that FITS file. A specific `BANK` is identified by an upper case letter from A through H and stored in the `BANK` keyword of the primary HDU.

Each spectrometer samples two input polarizations that can be further divided into as many as 8 frequency sub-bands. Full Stokes combinations are available in each sub-band although if the observing mode only requires self polarizations (no cross polarizations) then only those values will be saved in the fits file. Six binary table extensions will be produced for each `BANK` used for the observations: `SPURSPORT`, `STATE`, `SAMPLER`, `ACT_STATE` and `DATA`, and these tables will be placed in a single scan data FITS file for each scan. Since one FITS file will be created for each `BANK` a constraint on each bank output is that all spectra from all samplers and sub-bands for a given `BANK` will have the same number of frequency channels. Different spectrometers (`BANK`s) may produce spectra with different numbers of channels.

2 Primary HDU keywords

The VEGAS FITS keywords for the primary HDU conforms to the definition for common FITS headers as described in GBT Software Project Note 4.0 "Device and Log FITS Files for the GBT." Note that VEGAS, like the GBT Spectrometer, produces concurrently separate files for each `BANK` rather than collating them into one file. The FITS file primary HDU provides a summary of the file contents.

The primary header keywords are as follows:

```
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= 'VEGAS   '           / device or program of origin
GBTMCVER= 'vpm_reg_cand_mtx'   / telescope control software release
FITSVER = '1.2     '           / FITS definition version for this device
SIMULATE=                    0 / Is the instrument in simulate mode?
DATE-OBS= '2017-12-11T17:57:36' / Manager parameter startTime
TIMESYS = 'UTC     '           / time scale specification for DATE-OBS
TELESCOP= 'NRAO_GBT'           / Green Bank Telescope (Robert C. Byrd 100m)
OBJECT  = 'unknown '           / Manager parameter source
PROJID  = 'JUNK    '           / Manager parameter projectId
SCAN    = '174     '           / Manager parameter scanNumber
DATEBLD = '2017-12-01T22:12:28' / Date manager was built
DAQGVERS= 'SL_16.10.12_-27-g472857fc525a' / DAQ Git version hash
BANK    = 'B       '           / spectrometer identifier
NCHAN   =                32768 / number of channels in each spectrum
MODE    = '4       '           / VEGAS mode
SELFTEST=                    0 / Is VEGAS in self-test mode?
BASE_BW =              9.5E+02 / Base band bandwidth in MHz
NOISESRC= 'OFF     '           / Noise source, ON or OFF
BOFFILE = 'l1_ver137_1500_511_fir_2015_Dec_10_1529.bof'
ADCCONFH= '4d08b405b7859adf5734ff57139c8d6045a3d8f1 '
```

```
MMCM0   =                   40 / MMCM optimal phase for ADC 0
MMCM1   =                   36 / MMCM optimal phase for ADC 1
ADCSAMPF=      3.000000000E+09 / ADC Sampling frequency in Hz
COMMENT ***
COMMENT The following are VEGAS status shared memory keyword/value pairs
COMMENT ***

... shared keyword/values not show in this example ...
END
```

There are six non-standard keywords before the VEGAS shared memory section.

- `BANK` Identifies one of the eight possible banks (A through H) that the file is associated with.
- `MODE` A string used to describe the basic setup, used for engineering purposes.
- `DATEBLD` A string identifier which documents the build date of the vegas manager executable.
- `DAQGVERS` The git hash identifying the version of the data acquisition program.
- `SIMULATE` When non-zero this indicates that the VEGAS Manager is running in simulate mode and any data stored is generated by the software itself.
- `BASE_BW` The base band bandwidth. This filter determines the bandwidth of the base band.
- `NOISESRC` Is the noise source ”ON” or ”OFF”? This keyword will have one of those two values.
- `SELFTEST` When non-zero this indicates that VEGAS is running in self-test mode.
- `BOFFILE` The bof file in use for this scan.
- `ADCCONFH` The git hash of the ADC profile used to program the ADC.
- `MMCM0 and MMCM1` The values used to set the optimal phase for the two ADC inputs.
- `ADCSAMPF` The ADC sampling frequency used during data acquisition.

As for all GBT scan data FITS files, the `DATE-OBS` keyword takes the value of the Manager parameter `startTime`. This is the scheduled start time for the scan as a whole. The actual time at which an integration is initiated is recorded in the `DATA` table.

The keyword/value pairs at the end, after the BOFFILE keyword, are all VEGAS status shared memory keywords. It is useful to record these during commissioning. The specific status keywords may change over time without notice and readers should not rely on their presence in any VEGAS fits data. The specific VEGAS status shared memory keywords are not noted in this document.

3 SPURS Binary Table Extension

VEGAS data shows a consistent rail of spikes, which become more apparent in data with low signal. These spikes appear at regular and predictable intervals in single channels of the VEGAS data, and look like narrow band RFI. The SPURS table lists for each entry in the SAMPLER table, the predicted baseband frequencies and channel numbers where ADC spurs are predicted to occur.

The spurs have many possible origins, the most likely being phase mis-alignment of the four interleaved ADC’s, the FPGA clock etc. The center channel in all modes will have a strong spur.

The calculation of the spurs can be perform using the following formula:

```
SPUR_CHANNEL = (J*ADCSAMPF/64-CRVAL1)/CDELT1+CRPIX1
```

where J ranges from 0 to 32 inclusive.

The SPURS table for VEGAS will contain entries for each I/F listed in the SAMPLER table. The SPURS columns are:

SAMPLER This is an index (i.e. row number) into the SAMPLER table which the spur refers to.

SPURCHAN This is the channel number, starting at 1, where a spur is located.

SPURFREQ The baseband frequency (frequency at the vegas input port) of the spur.

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   16 / width of table in bytes
NAXIS2  =                    5 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    3 / number of fields in each row
TTYPE1  = 'SAMPLER '           / label for field   1
TFORM1  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT1  = 'INDEX   '           / physical unit of field
TTYPE2  = 'SPURCHAN'           / label for field   2
TFORM2  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT2  = 'INDEX   '           / physical unit of field
TTYPE3  = 'SPURFREQ'           / label for field   3
TFORM3  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT3  = 'HZ      '           / physical unit of field
EXTNAME = 'SPURS   '           / name of this binary table extension
COMMENT The known spurs are calculated from the formula
COMMENT SPUR_CHAN = (J*ADCSAMPF/64-CRVAL1)/CDELT1+CRPIX1
COMMENT Where J ranges from 0 to 32
COMMENT Spurs outside the bandpass are excluded from the table
COMMENT Note the center channel (CRPIX1) is always flagged as a spur
END
--------------------------------------------------------------------------------

SPURS Binary Tables Extension: Data (example)
--------------------------------------------------------------------------------

      SAMPLER     SPURCHAN       SPURFREQ
           1J           1J             1D
        INDEX        INDEX             Hz
1           1          513    7.50000E+08
2           2          513    7.50000E+08

--------------------------------------------------------------------------------
```

4 PORT Binary Table Extension

The VEGAS FITS keywords for the PORT table conforms to the definition for common FITS headers as described in GBT Software Project Note 4.0 “Device and Log FITS Files for the GBT.”

The `PORT` table for VEGAS will always contain 2 rows.

```
PORT Binary Tables Extension: Header
--------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   12 / width of table in bytes
NAXIS2  =                    2 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    4 / number of fields in each row
TTYPE1  = 'BANK    '           / label for field   1
TFORM1  = '1A      '           / data format of field: ASCII Character
TUNIT1  = 'INDEX   '           / physical unit of field
TTYPE2  = 'PORT    '           / label for field   2
TFORM2  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT2  = 'INDEX   '           / physical unit of field
TTYPE3  = 'MEASPWR '           / label for field   3
TFORM3  = '1E      '           / data format of field: 4-byte REAL
TUNIT3  = 'dBm     '           / physical unit of field
TTYPE4  = 'T_N_SW  '           / label for field   4
TFORM4  = '5A      '           / data format of field: 5 ASCII Characters
EXTNAME = 'PORT    '           / name of this binary table extension
END
--------------------------------------------------------------------------------

PORT Binary Tables Extension: Data (complete)
--------------------------------------------------------------------------------

      BANK     PORT       MEASPWR  T_N_SW
        1A       1I            1E      5A
     INDEX    INDEX           dBm
1        A        1 -1.973621E+00    TONE
2        A        2 -2.048295E+00    TONE

--------------------------------------------------------------------------------
```

The `PORT` columns are:

`BANK` This is itentical to the `BANK` keyword value in the primary HDU. It is included here as a column with a constant value to keep this FITS file format as similar to that found in the original GBT Spectrometer as possible.

`PORT` When used with `BANK` this identifies a unique row in the IF FITS for this scan and backend (VEGAS) associate with each input to this instrument.

`MEASPWR` The average measured power in the spectrometer for that port derived from the ADC histogram just prior to the start of the scan.

`T_N_SW` The TONE/NOISE switch value. This will be either 'TONE' or 'NOISE'.

5 STATE Binary Table Extension

The VEGAS FITS keywords for the `STATE` table conforms to the definition for common FITS headers as described in GBT Software Project Note 4.0 "Device and Log FITS Files for the GBT." To reiterate from that document (which describes the `STATE` table found in many GBT FITS files) for the `SIGREF` column, value 0 indicates signal and non-zero indicates some reference; and for the `CAL` column, value 1 indicates `CAL` signal was on and value 0 indicates `CAL` signal off. An example `STATE` header, plus ASCII listing of typical table data, is given below:

```
STATE Binary Tables Extension: Header
--------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   24 / width of table in bytes
NAXIS2  =                    4 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    4 / number of fields in each row
TTYPE1  = 'BLANKTIM'           / label for field   1
TFORM1  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT1  = 'SECONDS '           / physical unit of field
TTYPE2  = 'PHSESTRT'           / label for field   2
TFORM2  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT2  = 'NONE    '           / physical unit of field
TTYPE3  = 'SIGREF  '           / label for field   3
TFORM3  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT3  = 'T/F     '           / physical unit of field
TTYPE4  = 'CAL     '           / label for field   4
TFORM4  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT4  = 'T/F     '           / physical unit of field
EXTNAME = 'STATE   '           / name of this binary table extension
NUMPHASE=                    4 / Number of Phases if only Internal Switching Sig
SWPERIOD=            1.000E+00 / Switching period
MASTER  = 'VEGAS'              / Switching Signals Master
END
--------------------------------------------------------------------------------

STATE Binary Tables Extension: Data (complete)
--------------------------------------------------------------------------------
             BLANKTIM             PHSESTRT       SIGREF          CAL
                   1D                   1D           1J           1J
              SECONDS                 NONE          T/F          T/F
1  2.000000000000E-03   0.000000000000E+00            0            0
2  2.000000000000E-03   2.500000000000E-01            0            1
3  2.000000000000E-03   5.000000000000E-01            1            0
4  2.000000000000E-03   7.500000000000E-01            1            1
--------------------------------------------------------------------------------
```

Note that this table describes VEGAS's internal configuration of switching signals, it is named `STATE` in accordance with common M&C usage and convention in the other GBT FITS files. It is possible that VEGAS is being driven by external signals (or a combination of both), and so the actual switching signals in use are given in the `ACT_STATE` table described below.

Also note that VEGAS can make use of an additional switching state. Since that is not yet implemented across other devices that make use of the `STATE` table (e.g. the LO1 FITS file) the changes necessary to support that fully in this FITS file are intentionally omitted from the current version. The `ACT_STATE` table does have columns for all types of switching. In the interim, the `ESIGREF1` or `ISIGREF1` column in the `ACT_STATE` table is equivalent to the `SIGREF` column here (depending on whether internal or external switching is being used).

6 SAMPLER Binary Table Extension

The `SAMPLER` table describes the combinations of the `PORT`s found in the `DATA` table for this `BANK` as well as the sub-bands that are sampled for each combination. The order of the rows of the `SAMPLER` table define the order of the combinations and sub-bands for the `SAMPLER` axis of the `DATA` column in the `DATA` table (see below). An example header, plus ASCII listing of typical table data, is given below:

```
SAMPLER Binary Tables Extension: Header
--------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   36 / width of table in bytes
NAXIS2  =                    4 / 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
TTYPE1  = 'BANK_A'             / label for field   1
TFORM1  = '1A      '           / data format of field: 2-byte INTEGER
TUNIT1  = 'INDEX   '           / physical unit of field
TTYPE2  = 'PORT_A'             / label for field   2
TFORM2  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT2  = 'INDEX   '           / physical unit of field
TTYPE3  = 'BANK_B'             / label for field   3
TFORM3  = '1A      '           / data format of field: 2-byte INTEGER
TUNIT3  = 'INDEX   '           / physical unit of field
TTYPE4  = 'PORT_B'             / label for field   4
TFORM4  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT4  = 'INDEX   '           / physical unit of field
TTYPE5  = 'DATATYPE'           / label for field   4
TFORM5  = '4A      '           / data format of field: 2-byte INTEGER
TTYPE6  = 'SUBBAND '           / label for field 5
TFORM6  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT6  = 'INDEX   '           / physical unit of field
TTYPE7  = 'CRVAL1  '           / label for field   6
TFORM7  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT7  = 'Hz      '           / physical unit of field
TTYPE8  = 'CDELT1  '           / label for field   7
TFORM8  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT8  = 'Hz      '           / physical unit of field
TTYPE9  = 'FREQRES '           / label for field   8
TFORM9  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT9  = 'Hz      '           / physical unit of field
CRPIX1  =           5.1300E+02 / Reference channel
POLARIZE= 'CROSS'              / 'CROSS' implies poln measurement, else 'SELF'
```

```
EXTNAME = 'SAMPLER '           / name of this binary table extension
END
--------------------------------------------------------------------------------

SAMPLER Binary Tables Extension: Data (complete)
--------------------------------------------------------------------------------
   BANK_A  PORT_A  BANK_B  PORT_B  DATATYPE  SUBBAND
       1A      1I      1A      1I        4A       1I
    INDEX   INDEX   INDEX   INDEX              INDEX
1       A       1       A       1      REAL        0
2       A       2       A       2      REAL        0
3       A       1       A       2      REAL        0
4       A       1       A       2      IMAG        0

              CRVAL1              CDELT1             FREQRES
                  1D                  1D                  1D
                  Hz                  Hz                  Hz
1 2.180000000000E+09  1.464843750000E+06  1.464843750000E+06
2 2.180000000000E+09  1.464843750000E+06  1.464843750000E+06
3 2.180000000000E+09  1.464843750000E+06  1.464843750000E+06
4 2.180000000000E+09  1.464843750000E+06  1.464843750000E+06

--------------------------------------------------------------------------------
```

Each row of the `SAMPLER` table describes one combination of ports and sub-bands sampled by this bank of VEGAS. The total number of rows therefore defines the size of the second dimension (`TDESC2 = 'SAMPLER'`) of the (`TTYPE3 = 'DATA'`) column of the `DATA` table described below, , and the row number provides the index into this dimension.

In the `'CROSS' POLARIZE` mode there will be 4 rows for each sub-band and in the `'SELF' POLARIZE` mode there will be 2 for each sub-band. Since the `DATA` column of the `DATA` table does not hold complex numbers, one of the cross-polarization `SAMPLER`s corresponds to the real part of the complex cross-polarization spectrum and the other cross-polarization `SAMPLER` corresponds to the imaginary part.

There are either 1 or 8 sub-bands in a VEGAS bank.

The definition of the `SAMPLER` columns are as follows:

`BANK_A` This will always be the same as the `BANK` keyword in the Primary HDU. It appears here as a column so that this table closely resembles the same named table in the original GBT Spectrometer FITS files.

`PORT_A` The `PORT` corresponding to the "A" input. This will usually be the same `PORT` found in the first row of the `PORT` table.

`BANK_B` This will always be the same as the `BANK` keyword in the Primary HDU.

`PORT_A` The `PORT` corresponding to the "B" input. This will usually be the same as the `PORT` found in the second row of the `PORT` table.

`DATATYPE` One of "REAL" or "IMAG". These describe the type of data found at each element along the `SAMPLER` axis of the `DATA` . For self-polarizations (`PORT_A` equal to `PORT_B'`) this is always "REAL". When cross-polarizations are involved the same value of the two ports will appear in two rows with the real part associated with one row and the imaginary part associated with the other row of this table.

`SUBBAND` An index (counting from 0) indicating which sub-band this `SAMPLER` corresponds to.

`CRVAL1` The frequency at the center of the channel given by the `CRPIX1` keyword, in Hz.

`CDELT1` The increment between adjacent channels, in Hz. Note that the total bandwidth for a given `SAMPLER` is always the absolute value of `CDELT` times the number of channels.

`FREQRES` The effective channel resolution, in Hz. This value is always positive.

The definitions of the `SAMPLER` table-specific keywords are as follows:

`CRPIX1` The location of the reference channel corresponding to `CRVAL1` for all sub-bands. Note that this follows the FITS convention where channels are numbered from 1 to NCHAN.

`POLARIZE` String describing the polarization combinations present. Values are `'SELF'` or `'CROSS'` polarizations.

The center IF frequency of each channel, i, is given by: $CRVAL1 + CDELT1 \times (CRPIX1 - i)$

This value should be used along with appropriate values from the IF FITS file and the LO1 FITS file in the Sky Frequency Formula, which is described in the IF FITS file documentation.

7 ACT_STATE Binary Table Extension

The spectrometer `ACT_STATE` table describes the state of the calibration and switching signals input to the spectrometer for each element along the `ACT_STATE` axis of the `DATA` column in the `DATA` binary table extension (see below). The order of the rows here is equivalent to the pixel ordering along the `ACT_STATE` axis.

An example `ACT_STATE` header, plus ASCII listing of typical table data, is given below:

```
ACT_STATE Binary Table Extension: Header
-------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   24 / width of table in bytes
NAXIS2  =                    4 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    6 / number of fields in each row
TTYPE1  = 'ISIGREF1'           / label for field   1
TFORM1  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT1  = 'T/F     '           / physical unit of field
TTYPE2  = 'ISIGREF2'           / label for field   2
TFORM2  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT2  = 'T/F     '           / physical unit of field
TTYPE3  = 'ICAL    '           / label for field   3
TFORM3  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT3  = 'T/F     '           / physical unit of field
TTYPE4  = 'ESIGREF1'           / label for field   4
TFORM4  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT4  = 'T/F     '           / physical unit of field
TTYPE5  = 'ESIGREF2'           / label for field   5
TFORM5  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT5  = 'T/F     '           / physical unit of field
TTYPE6  = 'ECAL    '           / label for field   6
TFORM6  = '1J      '           / data format of field: 4-byte INTEGER
```

```
TUNIT6  = 'T/F     '           / physical unit of field
EXTNAME = 'ACT_STATE'          / name of this binary table extension
END
--------------------------------------------------------------------------------

ACT_STATE Binary Tables Extension: Data (complete)
--------------------------------------------------------------------------------
      ISIGREF1    ISIGREF2        ICAL    ESIGREF1    ESIGREF2        ECAL
            1J          1J          1J          1J          1J          1J
           T/F         T/F         T/F         T/F         T/F         T/F
1            0           0           0           0           0           0
2            0           0           1           0           0           0
3            1           0           0           0           0           0
4            1           0           1           0           0           0
--------------------------------------------------------------------------------
```

There can be different numbers of active columns in the ACT_STATE tables corresponding to the number of switching signals selected for the bank during a scan. The keyword NAXIS2 is the number of rows in the table and defines the number of states of the spectrometer during an integration. The number of rows in the ACT_STATE table is a power of 2, where the power is the number of active columns of the table. For example if there are 3 switching signals used, then there will be 3 columns in the table whose values are changing from row to row resulting in 8 rows.

The inputs to the spectrometer are arranged into two groups, internal and external inputs.

Note: VEGAS can be configured with 2 independent SIGREF *switching states. Currently only one of these can be set in the* STATE *table and so the* ?SIGREF2 *columns shown here are intended for future use.* ?SIGREF1 *is equivalent to the* SIGREF *column in the* STATE *table. Here "?" can be either "I" or "E" for internal or external switching signals.*

The definitions of the possible ACT_STATE table columns are as follows:

ISIGREF1 Internal primary Signal-Reference indicator. Value zero indicates the signal source, a non-zero value indicates a reference source.

ISIGREF2 Internal secondary Signal-Reference indicator.

ICAL Internal indication of the state of switching CAL noise diode. Value one indicates CAL signal was on. Value zero indicates the CAL signal off.

Identical columns labelled ESIGREF1, ESIGREF2 and ECAL are used for the external switching signals.

8 DATA Binary Table Extension

The DATA table contains a (large) DATA column containing data from this BANK of VEGAS.

An example extension header, and a subset of the data is given below:

```
DATA Binary Tables Extension: Header
--------------------------------------------------------------------------------
TENSION= 'BINTABLE'            / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                65676 / width of table in bytes
```

```
NAXIS2  =                    2 / 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
TTYPE1  = 'DMJD    '           / label for field   1
TFORM1  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT1  = 'd       '           / physical unit of field
TTYPE2  = 'INTEGRAT'           / label for field   2
TFORM2  = '16E     '           / data format of field: 4-byte REAL
TUNIT2  = 'sec     '           / physical unit of field
TDIM2   = '(4,4)   '           / size of the multidimensional array
TDESC2  = 'SAMPLER,ACT_STATE'  / definition of axes
TTYPE3  = 'DATA    '           / label for field   3
TFORM3  = '16384E  '           / data format of field: 4-byte REAL
TUNIT3  = 'COUNTS  '           / physical unit of field
TDIM3   = '(1024,4,4)'         / size of the multidimensional array
TDESC3  = 'CHAN,SAMPLER,ACT_STATE'  / definition of axes
TTYPE4  = 'UTCDELTA'           / label for field   4
TFORM4  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT4  = 's       '           / physical unit of field
TTYPE5  = 'INTEGNUM'           / label for field   5
TFORM5  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT5  = 'id      '           / physical unit of field
TTYPE6  = 'ACCUMID '           / label for field   6
TFORM6  = '4J      '           / data format of field: 4-byte INTEGER
TUNIT6  = 'swsigbit'           / physical unit of field
TTYPE7  = 'STTSPEC '           / label for field   7
TFORM7  = '4J      '           / data format of field: 4-byte INTEGER
TUNIT7  = 'SPCOUNT '           / physical unit of field
TTYPE8  = 'STPSPEC '           / label for field   8
TFORM8  = '4J      '           / data format of field: 4-byte INTEGER
TUNIT8  = 'SPCOUNT '           / physical unit of field
TTYPE9  = 'TIME_CTR'           / label for field   9
TFORM9  = '1K      '           / data format of field: 8-byte INTEGER
TUNIT9  = 'FPGACLKS'           / physical unit of field
SWPERINT=                    2 / Number of switching periods per integration
UTCSTART= 5.8672000000000000E+04 / Actual start time in seconds since midnight
UTDSTART=                56526 / Actual start time in MJD
DURATION=        2.0000000E+00 / Length of one integration, seconds.
EXTNAME = 'DATA    '           / name of this binary table extension
END
--------------------------------------------------------------------------------

DATA Binary Tables Extension: Data (1 partial row)
--------------------------------------------------------------------------------
                 DMJD       INTEGRAT          DATA          UTCDELTA
                   1D           128E       524288E                1D
                    d            sec        COUNTS                 s
1    5.652667908568E+04  9.977110E-01  4.181088E+09  1.001973168000E+00

   INTEGNUM    ACCUMID    STTSPEC         STPSPEC      TIME_CTR
         1J         4J         4J              4J            1K
         id   swsigbit    SPCOUNT         SPCOUNT      FPGACLKS
1         0   (3,2,1,0) (4,1911,...) (1906,3813,...) 187869969
```

The DATA table contains rows describing the spectra obtained during the scan. Both the INTEGRAT and DATA columns contain multi-dimensional entries where the specific size and shape can vary from scan to scan (and from bank to bank). Within one BANK for one scan all spectra must have the same number of channels. The INTEGRAT column is column 2, and the structure of each 2 dimensional cell in that column is specified by the keyword TDIM2. The DATA column is column 3, and the structure of each 3 dimensional cell in that column is specified by the keyword TDIM3. For example, in a typical cross-polarization observation with 1024 channels, 2 sub-bands, and a switched noise cal, but no signal/reference switching then DATA has shape TDIM3='(1024,8,2)'. The INTEGRAT column for this observation would have TDIM2='(8,2)'.

The mid-point of a given integration is them found by adding half the integration duration (DURATION keyword value) to the DMJD start time of that integration.

Note: If the PRIMARY header keyword 'NORMALZD' is non-zero, then the DATA column values will be normalized by the appropriate INTEGRAT values. If the 'NORMALZD' is zero then the data should be divided by the appropriate INTEGRAT value before using them. The default is to normalize the data.

The table FITS keywords are described below:

SWPERINT Number of switching periods per integration. Note that the number of integrations (or rows in the DATA table) per scan is given by NAXIS2.

UTCSTART and UTDSTART Exact start time of the scan, should differ from DATE-OBS only for use in pulsar mode.

DURATION The integration duration, in seconds, including any blanking time. This is the total clock time spent on that integration.

The columns of the table are defined as follows:

DMJD The date and time of the start of the integration (double modified Julian day including fractional day).

INTEGRAT The data integration times in each of the states and samplers. The total number of elements in each cell of this column is $N_{sampler} \times N_{actstate}$.

DATA A three dimensional array of data values which may correspond to the real or imaginary part of a spectrum (see the SAMPLER table for additional details on the data type stored here). The size of the array in each cell is $N_{chan} \times N_{sampler} \times N_{actstate}$.

UTCDELTA The offset, in seconds, from the UTCSTART keyword value found in this HDU to the start of the integration. This column is present for users who require extra time precision than is available in the DMJD column. The MJD time at the start of the integration from $UTDSTART + (UTCSTART + UTCDELTA)/86400.0$ should equal DMJD to the accuracy of the DMJD value.

TDIM2 The dimensions of the INTEGRAT column entry are described by this keyword. The first dimension always has only one element.

TDIM3 The dimensions of the DATA column entry are described by this keyword.

TDESC2 Type of data along the axes of the 2 dimensional table element in the second column INTEGRAT

TDESC3 Type of data along the axes of the 3 dimensional table element in the third column DATA.

The INTEGNUM, ACCUMID, STTSPEC, STPSEC, and TIME_CTR columns are useful when debugging VEGAS. These are not generally useful during routine VEGAS observations and their values are not required when intrepeting and using the rest of the VEGAS FITS file contents.

Note: The TIME_CTR column has a type of "1K" which is a a later addition to the set of recognized binary table column types. It may be necessary to use the latest FITS software to be able to read this table, even if access to this specific column is not needed.