AIPS Memo 117 revised

AIPS FITS File Format

Eric W. Greisen

March 5, 2020

Contents

Abstract

AIPS has been writing images and *uv* data in FITS-format files for a very long time. While these files have been used widely in the community, there is a perception that a detailed document in still required. This memo is an attempt to meet that perception. *AIPS* FITS files for *uv* are conventions layered upon the standard FITS format to assist in the interchange of data recorded by interferometric telescopes, particularly by radio telescopes such as the EVLA and VLBA.

1 Introduction

Basic FITS was initially proposed by Wells, Greisen, and Harten 1981 [25] to enable transmission of image data between observatories. Greisen and Harten realized immediately that a similar mechanism was needed to transmit interferometer visibility data between, for example, Westerbork and the VLA. This led to the paper on "random groups" format (Greisen & Harten 1981 [11]), a somewhat kludgy extension to the initial format. The "kludge" was made necessary by the "*once FITS, always FITS*" philosophy that was adopted at that time (and remains true to this day). In these early days, it was realized that data other than the basic images and visibilities would also have to be transmitted. Examples are Clean Component tables and Antenna tables. Experiments along these lines led to the generalized extensions paper (Grosbøl, Harten, Greisen, Wells 1988 [12]) which described how the now IAU-standard FITS formats could be extended. Responding to needs within the *AIPS* software, Cotton, Tody, and Pence 1995 [3] eventually published the binary tables extension paper although the binary tables had been in use in *AIPS* for many years by that time.

Papers initially proposing format standards have to be written with sections explaining and defending the decisions made, worked examples, and other text inappropriate for a true standards document. This led NASA's Office of Standards Technology to form a working group to develop a carefully worded document to define the standard. That working group and its successors have developed a "Definition" document which is now at version 3.0 (Pence et al. 2010 [21]). All documents, and everything else you might wish to know about FITS are available at the web site maintained by Goddard Space Flight Center (`fits.gsfc.nasa.gov`).

At the recommendation of Commission 5, chaired by Bernard Hauk and Gart Westerhout, the IAU adopted the two initial papers as IAU standards (IAU 1982 [16]). Some years later a Working Group on FITS was created (IAU 1988 [17]) which is to the present time the ultimate authority on FITS matters.

AIPS was "born" July 1, 1979, three months after the FITS format was invented. It is then natural that the internal data structures of *AIPS* are extensions of the initial FITS data structures, while later FITS data structures were driven by the needs of *AIPS* and other software packages of the time. *AIPS* *uv* data have been backed up to tape (now disk) in random-groups format by the task `FITTP`. The associated table data are now added to the random-groups file by `FITTP` in binary-tables format. The random groups data structure has apparently felt unnatural to most of the FITS community. Therefore, it has been "deprecated," meaning that no new applications for it should be developed although existing applications, especially interferometry, are to be supported. For this reason, among others, the FITS IDI Convention (Greisen 2011 [8]), developed during the 1990s, made the choice to write *uv* data as a binary table, rather than random groups. In fact, the actual binary data of the random groups and binary table formats may be (and usually are) identical. What differs is the text headers describing the data. FITS-IDI was created to transmit raw data from correlators to data reduction software systems, so *AIPS* never developed the ability to write conforming FITS-IDI format. Instead, `FITAB` was developed to write the visibility data as a binary table which follows all of the associated binary tables in the FITS file. That order was chosen since visibility data are useless without the essential (and small) tables giving antenna locations, frequency information, etc. Random groups requires the visibility data to come first; so a premature end-of-file would cause the small but essential tables to be lost rather than some useful, but not essential, visibility data.

In the interests of keeping this document to a reasonable length, the reference chapters do not make any specific mention of any elements that can be inferred to be present from the requirements that *AIPS* FITS files be valid FITS files, as defined by version 3.0 of the standard, unless they have some additional meaning in the context of an *AIPS* file (*e.g.*, NAXIS values in tables). Although they are omitted from this document, these elements should be taken to be mandatory in *AIPS* FITS files. Appendix A is included to provide a primer on FITS for those not wishing to read the original papers and definition documents.

Character strings that should appear in *AIPS* FITS files exactly as they are written will be presented in a `typewriter-like` font. This font will also be used for the names of computer programs. Character-string values for FITS header keywords will be marked with single quotation marks, as in `'a character string'`; the quotation marks do not form a part of the string value but are required delimiters. Some keywords used in FITS files consist of a fixed portion followed by an integer suffix that may be different in different context. These will be indicated like `NAXIS`*n*, where the *n* denotes the integer suffix. Parameters that may have different values under different circumstances are denoted in *italic font*.

Each keyword in a FITS header is associated with a value that has a specific type. In this document, these types are denoted by the letters shown in Table 1.

TABLE 1: Type codes for keyword values

Code	Type
I	integer number
L	logical
A	character string (usually $\leq$ 8 characters
E	floating-point number (usually double precision)
D	date string

A date is a character string in one of two specific formats. The first format is `'`*DD*`/`*MM*`/`*YY*`'`, where *DD* is a two-digit day number, *MM* is a two-digit month number, and *YY* is a two-digit year number, suitable for use during the twentieth century. The preferred format is `'`*YYYY*`-`*MM*`-`*DD*`'`, where *YYYY* is a four-digit year number suitable for use in any recent century. Although the FITS standard allows times to be appended to the second form of a date string, times will not be appended to date strings in *AIPS* FITS files.

Each column in a FITS binary table has a type which denotes the kind of values that may appear in that column. Each column holds a one-dimensional array of some base type with a fixed number of elements. The base type of an array is denoted by a single-character code in this document. These codes correspond to those used for the `TFORM`*n* values in the table header and are listed in Table 2.

TABLE 2: Basic types for fields in binary tables

Code	Type
L	logical
I	16-bit integer
J	32-bit integer
A	character
E	32-bit floating-point number
D	64-bit floating-point number

In the simplest case, the number of elements in the array is given as a repeat count preceding the code for the basic type, *e.g.*, `4J` for an array of four 32-bit integers. Some fields are, however, considered to be multi-dimensional arrays in *AIPS* FITS tables despite being declared as one-dimensional arrays in the FITS headers. In these cases, the array dimensions will appear in parentheses following the basic type, *e.g.*, `E(4, 32)` for a two-dimensional array with 4 columns and 32 rows. In the table header, the repeat count shall be the product of all the dimensions and the data in the array shall be laid out so that the index of the first dimension varies fastest, followed by the second dimension, and so on. *AIPS* FITS does use the multi-dimensional array convention of the FITS standard and programs that read *AIPS* FITS files can rely on the presence of `TDIM`*n* keywords. The order of columns in a table should not matter.

2 Image data

The FITS format for image data is relatively simple. The file begins with a header containing 36 80-character fields per logical record. The image data are an array of binary values immediately following the header, fully packed in 2880 bytes per logical record. The image is viewed as a Fortran array $A(n_1, n_2, n_3)$ where the indices n_i run from 1 to N_i where N_i is a positive integer. Index n_1 varies fastest in the sequence of values, n_2 second fastest, and so on. The details were spelled out by Wells, Greisen, and Harten 1981 [25] and clarified in the FITS definition (Pence, *et al.* 2010 [21]).

An example of the primary HDU of an image FITS file from *AIPS* is shown in Appendix B.1 on page 81. Tables accompanying images are described in Section 6 on page 57.

3 Visibility data

3.1 Random groups form

A group in the random groups format is a list of parameter values plus an array of data. The array is of any number of dimensions and the values are taken as falling on a regular grid in either true or conventional coordinate systems. The parameters represent, in general, coordinates that describe the array. A simple example would be a small, 2-dimensional sky image accompanied by parameters giving the central right ascension and declination plus ones giving the observing frequency and time. A random groups FITS file then consists of some number of such groups, each with the same number and type of parameters and the same number and type of axes including the same number of values on each axis.

The "kludge" added to maintain once FITS, always FITS for random groups was to add 1 to the dimension of the array and to define that the number of points on the first axis is zero. This tells readers of FITS images that there is no regular image. Random groups readers then use axes 2-`NAXIS` as the actual axes of the image.

An example of the primary HDU of a random groups "UVFITS" file is shown in Section B.2 on page 82.

3.1.1 Regular axes

The regular axes of the data array in *AIPS* *uv* data sets are listed in Table 3.

TABLE 3: Regular axes for the data array

Name	Mandatory ?	Description
`COMPLEX`	yes	Real, imaginary, weight
`STOKES`	yes	Stokes parameter
`FREQ`	yes	Frequency (spectral channel)
`IF`	no	Spectral window number
`RA`	yes	Right ascension of the phase center
`DEC`	yes	Declination of the phase center

The `COMPLEX` axis shall be the first (*i.e.*, the fastest changing) axis in the data matrix. It shall have a `NAXIS2` value of 2 or 3 and `CDELT2`, `CRPIX2`, and `CRVAL2` shall all have the value 1.0. The first entry on this axis contains the real part of a complex visibility and the second contains the corresponding imaginary component. If a third element is present, then this shall contain the weight for this visibility measurement. Weights $\leq$ 0 indicate that the visibility measurement is flagged and that the values may not be in any way meaningful.

The `STOKES` axis enumerates polarization combinations. The corresponding `NAXIS`m value, denoted n_{stokes}, shall be no less than 1 and no greater than 4. The `CRPIX`m value shall be 1.0. Stokes is a conventional axis:

values of 1 through 4 are assigned to Stokes I, Q, U, and V, values of -1 through -4 are assigned to RR, LL, RL, and LR polarization products, respectively. Values -5 through -8 are assigned to XX, YY, XY, and YX polarization products, respectively.

The `FREQ` axis enumerates frequency channels. The corresponding `CRVAL`m shall have the reference frequency for the data set. Conventionally, it is the frequency for the first IF but that is not required. Both `CRVAL`m and `CDELT`m are given in Hz. The number of spectral channel will be denoted by n_{chan}.

The `IF` axis enumerates spectral windows (frequency bands). The corresponding `CRVAL`m, `CRPIX`m, and `CDELT`m keywords shall all have the value 1.0. The `IF` axis may be omitted if and only if there is only one band and there is only one frequency setup. If this axis is present, the FITS file must include an `AIPS FQ` binary table (Section 4.7 on page 23). The number of pixels on the `IF` axis will be denoted by n_{IF}.

The `RA` and `DEC` axes shall both have the corresponding `nAXIS`m values of 1. If only one source is present in the file and no `SOURCE` tables are present, then the `CRVAL`m keyword for the `RA` axis shall give the right ascension of the phase center in degrees and the `CRVAL`m keyword for the `DEC` axis shall give the declination of the phase center in degrees. These coordinates shall be those of the standard equinox and that standard equinox shall be specified in the header. If more than one source is present in the file, then the `CRVAL`m keywords for both the `RA` and the `DEC` axes shall have the value 0.0 and no equinox need be specified.

3.1.2 Random parameters

The name of each random parameter is given as the value of the corresponding `PTYPE`n keyword, where n is the parameter number from 1 through `PCOUNT`. In tables, the parameter name is specified with `TTYPE`n keyword, where n is the column number in which the value of the random parameter appears. The recognized values are listed in Table 4.

TABLE 4: Random parameter names

Name	Type	Units	Description
`UU--`	1E	seconds	u baseline coordinate
`VV--`	1E	seconds	v baseline coordinate
`WW--`	1E	seconds	w baseline coordinate
`DATE`	2E	days	Julian date
`BASELINE`	1E		Baseline number
`SOURCE`	1E		Source ID number
`INTTIM`	1E	seconds	Integration time
`FREQSEL`	1E		Frequency setup ID number
`CORR-ID`	1E		VLBA specific
`GATEID`	1E		VLBA specific
`FILTER`	1E		VLBA specific
`SUBARRAY`	1E		Subarray number
`ANTENNA1`	1E		Lower antenna number
`ANTENNA2`	1E		Higher antenna number

Baseline coordinates. Three of the random parameters shall be used to specify the baseline coordinates for the visibility measurements in light seconds. The three coordinates are designated by names that begin with `UU`, `VV`, and `WW`, which correspond to the u, v, and w coordinates at the coordinate equinox. The first two letters may be followed by an optional suffix that indicates the coordinate system used for the baseline coordinates. If the suffix is omitted, then the `---SIN` convention is assumed. The suffixes must match on all three baseline coordinate parameters.

If the suffix is `---SIN`, then the w axis lies along the line of sight to the source and the u and v axes lie in a plane perpendicular to the line of sight with v increasing to the north and u increasing to the east. If the suffix is `---NCP`, then the w axis points to the north pole, the v axis is parallel to the projection of the line of sight into the equator with the v coordinate increasing away from the source and the u coordinate completes the right-handed Cartesian triad (u, v, w). Note that the `---NCP` system is normally used only with East-West interferometers in which the value of w is zero.

`FITTP` uses the `PSCALn` parameters for the baseline coordinates to encode the translation between the required units (seconds) and *AIPS* internal units (wavelengths). Similarly, `FITAB` uses the `TSCALn` parameters to encode this translation.

DATE. Two random parameters, both named `DATE`, shall be used to record the time at which the visibility measurements in a record were taken. Following conventions in the random groups format, the value of the (double-precision) sum of the parameters shall be the Julian date of the measurement. The time so recorded shall be the central time in the integration period and shall also be the time at which the (u, v, w) coordinates are valid.

`FITTP` uses the `PZERO`n parameter of the first of the two `DATE` parameters to encode the Julian date at midnight of the first day of the observation. As with the u, v, w, this means that the actual parameter values, before scaling and offset, are taken directly from the *AIPS* internal values.

Integration time. The length of the period over which the data were integrated may optionally be supplied in seconds as the value of the `INTTIM` parameter.

Baseline specification. The baseline (telescope pair) from which the data were obtained shall be formed by multiplying the number of the first antenna by 256 and then adding the number of the second antenna. The subarray number is encoded by adding $0.01\times$ (subarray number -1). Alternatively, these three parameters may be given by three separate random parameters. This new form will be written by *AIPS* to FITS files only if the maximum antenna number is greater than 255. Only one of the two forms of baseline identification may be used.

Source identification number. If the file contains observations of more than one source, then the identification number of the source being observed shall be given as the value of the `SOURCE` parameter. If this parameter is present, the FITS file must include an `AIPS SU` (Section 4.17 on page 35) binary table.

Frequency setup number. If the file contains observations made using more than one frequency setup, then the identification of the frequency setup that was used shall be recorded as the value of the `FREQSEL` parameter. If this parameter is present, the FITS file must include an `AIPS FQ` (Section 4.7 on page 23) binary table.

The frequency setup number has been a source of much confusion. If, at some time, data are taken with a given set of spectral windows and at a different time, data are taken at a different set of spectral windows, then these two data types are differentiated by assigning to them different frequency setup numbers. The frequency setup to which a particular group applies is then indicated with the `FREQSEL` random parameter. Note that two frequency setups may be included in the same FITS file only if they have the same number of spectral windows (those frequencies observed at the same time) and the same number of polarizations and spectral channels. If they do not have an identical array structure, then they must appear in separate FITS files. Note that the frequencies of the IFs and the frequency separations of each frequency setup may differ, but there must be the same number of IFs and same number of spectral channels. The polarization values for all frequency setups within a FITS file must be identical.

3.2 UV-tables form

In the UV-tables form, the visibility data are written as a FITS binary table, normally placed after the other table extensions. The primary HDU has an *AIPS* conventional form meant primarily to be so odd as to act as a reliable identifier. The primary HDU asserts that the primary data has two axes, the first of which has 777777701 values while the second has zero values. This is sufficient to tell all FITS readers that the primary data set is not a random groups data set and otherwise contains no data. An example primary HDU is shown in Section B.3 on page 84

The actual table gives a number of standard and non-standard FITS keywords in its header to convey information often conveyed in the primary HDU. Table 5 (page 10) lists the more interesting of these. In this, and all following, keyword tables, the keywords above the horizontal line are required and those below the line are optional. `HISTORY` and `COMMENT` cards are allowed. These are used to give the order in which the *uv* are sorted (actually columns on which the rows are sorted), the *AIPS* file name, and the "piece" number of the table along with the total number of pieces and the visibility numbers contained within the current piece. `FITAB` offers the option of writing multiple FITS files each containing a piece of the data. Each piece contains the full contents of any table not having a time axis and the appropriate portion of each table having a time axis. This allows for smaller FITS files, defending against media errors and assisting in network file transmission.

The table form used for *uv* data assigns a column to each of the random parameters and a column to the data matrix. Each row of the table is then a separate "group" in the notation of the random groups format. Tables do allow different digital formats in different columns (see *e.g.*, Table 2, page 6), but *AIPS* only uses this to transmit "compressed" visibility data in which the visibilities are scaled 16-bit integers while the scale and data weight for each row are random parameter columns. Table 6 shows the mandatory and optional columns of the `AIPS UV` table, while Table 7 shows the additional and changed columns for compressed data.

An example `AIPS UV` table header is shown in Appendix B.6 on page 107.

TABLE 5: Keywords for `AIPS UV` headers

Keyword	Value type	Value
`EXTNAME`	A	`'AIPS UV'`
`OBJECT`	A	Source name
`TELESCOP`	A	Telescope name
`INSTRUME`	A	Instrument name (receiver or ?)
`DATE-OBS`	A	Observation date
`DATE-MAP`	A	File processing date
`BSCALE`	E	1.0
`BZERO`	E	0.0
`BUNIT`	A	units, usually `'UNCALIB'` or `'JY'`
`EQUINOX`	E	Equinox of source coordinates and *uvw*
`ALTRPIX`	E	Reference pixel for velocity

TABLE 6: Mandatory and optional columns of the `AIPS UV` table

Name	Type	Units	Description
`UU--`	`1E`	seconds	u baseline coordinate
`VV--`	`1E`	seconds	v baseline coordinate
`WW--`	`1E`	seconds	w baseline coordinate
`DATE`	`1E`	days	Julian date
`BASELINE`	`1E`		Baseline number
`SOURCE`	`1E`		Source ID number
`FREQSEL`	`1E`		Frequency setup ID number
`VISIBILITIES`	n`E`	Jy	Fringe visibility data
`SUBARRAY`	`1E`		Subarray number
`ANTENNA1`	`1E`		Lower antenna number
`ANTENNA2`	`1E`		Higher antenna number
`INTTIM`	`1E`	seconds	Integration time
`CORR-ID`	`1E`		VLBA specific
`GATEID`	`1E`		VLBA specific
`FILTER`	`1E`		VLBA specific

Columns generally have the same meaning as described for the random parameters (Section 3.1.2 on page 8). Differences are indicated below.

DATE. One column named `DATE` shall be used to record the time at which the visibility measurements in a record were taken as a Julian date. The time so recorded shall be the central time in the integration period and shall also be the time at which the (u, v, w) coordinates are valid. `FITAB` uses the `TZERO`n keyword to encode the Julian date at midnight of the first day of the observation. As with the u, v, w, this means that the actual parameter values, before scaling and offset, are taken directly from the *AIPS* internal values.

SOURCE. One column shall be used to record the source number. The column is mandatory when there are data present from more than one pointing and/or phase stopping position. The column is optional when data from only one position is included. If this column is present, the FITS file must include an `AIPS SU` (Section 4.17 on page 35) binary table.

FREQSEL. One column shall be used to record the frequency setup number. The column is mandatory only when there are data present in the FITS file having more than one frequency setup number. If this column is present, the FITS file must include an `AIPS FQ` (Section 4.7 on page 23) binary table with rows describing each of the frequency setup numbers present in the data. Note that an `AIPS FQ` table is required if there are multiple spectral windows even when there is only one frequency setup number.

VISIBILITIES. One column shall be used to hold the fringe visibility data array. The `TDIM`n keyword shall be used to encode the pixel dimensions of this array. The coordinates of the array shall be encoded in the m`CRVL`n, m`CDLT`n, m`CRPX`n, AND m`CROT`n keywords, where m is the axis number within the array and n is the column number used for the array. This binary table coordinate convention is described in Greisen and Calabretta 2002 [9]. Units are Jy for the visibility real and imaginary parts and Jy^{-2} for the weight part of the `COMPLEX` axis.

An example `AIPS UV` table header for compressed data is shown in Appendix B.7 on page 109.

TABLE 7: Mandatory and changed columns of the `AIPS UV` table for compressed data

Name	Type	Units	Description
`WEIGHT`	`1E`	Jy^{-2}	Data weight for all vis.
`SCALE`	`1E`		Multiplier of visibilities
`VISIBILITIES`	m`I`	Jy	Fringe visibility data

WEIGHT. A single weight in Jy^{-2} is taken for all spectral windows, spectral channels, and polarizations in the visibility data array. Clearly this is not an accurate reflection of reality, but is needed to achieve the factor of 3 reduction in space possible with compressed data.

SCALE. The data in Jy are found by multiplying the recorded integers by the contents of the `SCALE` column. If there is a wide dynamic range in the visibilities of the row, then there is a loss of accuracy in this convention.

VISIBILITIES. The visibility data are recorded as above except that the array contains only the real and imaginary parts and these parts are recorded as 16-bit integers to be multiplied by the contains of the `SCALE` column and weighted by the contents of the `WEIGHT` column. Visibilities that are flagged are indicated by the value -32767, which appears as the value of the `TNULL`n keyword, where n is the column number of the `VISIBILITIES` column.

4 Tables initially from the correlator and real-time system

4.1 AIPS AN antenna table

The AIPS AN or antenna table contains information about the antennæ used in a FITS file. These characteristics include the polarization properties of the feeds at the level of IFs. Table 8 lists the keywords expected in the table header in addition to those required by the FITS format. This is an essential file although the details of antenna position are really only used by UVFIX and the polarization calibration is only used when applying IF-based D-term corrections. An example AIPS AN table header is shown in Appendix B.4.1 on page 84.

TABLE 8: Mandatory keywords for AIPS AN table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS AN'
EXTVER	I	Subarray number
ARRAYX	E	x coordinate of array center (meters)
ARRAYY	E	y coordinate of array center (meters)
ARRAYZ	E	z coordinate of array center (meters)
GSTIA0	E	GST at 0h on reference date (degrees)
DEGPDY	E	Earth's rotation rate (degrees/day)
FREQ	E	Reference frequency (Hz)
RDATE	D	Reference date
POLARX	E	x coordinate of North Pole (arc seconds)
POLARY	E	y coordinate of North Pole (arc seconds)
UT1UTC	E	UT1 - UTC (sec)
DATUTC	E	time system - UTC (sec)
TIMESYS	A	Time system
ARRNAM	A	Array name
XYZHAND	A	Handedness of station coordinates
FRAME	A	Coordinate frame
NUMORB	I	Number orbital parameters in table (n_{orb})
NO_IF	I	Number IFs (n_{IF})
NOPCAL	I	Number of polarization calibration values / IF (n_{pcal})
POLTYPE	A	Type of polarization calibration
FREQID	I	Frequency setup number

Array center. The ARRAYX, ARRAYY, and ARRAYZ keywords shall give the coordinates of the array center in the coordinate frame specified by the FRAME keyword. Antenna coordinates in the main part of the table are given relative to the array center, but rotated to the longitude of the array center.

GST at midnight. The value of the GSTIA0 keyword shall be the Greenwich sidereal time in degrees at zero hours on the reference date for the array in the time system specified by the TIMESYS keyword.

Earth rotation rate. The value of the DEGPDY keyword shall be the rotation rate of the Earth in degrees per day on the reference date for the array.

Reference date. The value of the RDATE parameter will be the date for which the time system parameters GSTIA0, DECPDY, and IATUTC apply. If the table contains orbital parameters for orbiting antennæ, this keyword also designates the epoch for the orbital parameters.

Frequency. The value of the FREQ keyword and the FREQID keyword give the reference frequency in Hz and the frequency setup number for the subarray to which this antenna table applies.

Reference date. The value of the RDATE parameter will be the date for which the time system parameters GSTIA0, DECPDY, and IATUTC apply. If the table contains orbital parameters for orbiting antennæ, this keyword also designates the epoch for the orbital parameters.

Polar position. The values of the POLARX and POLARY keywords shall give the x and y offsets of the North Pole in arc seconds on the reference date for the array with respect to the coordinate system specified by the FRAME keyword. The units were changed from the meters specified by the earlier documents, but seldom used in actual implementations. Note that arc seconds and meters can be told apart, at least in recent decades. If $\sqrt{P_x^2 + P_y^2} < 0.6$, the units are arc seconds.

Difference between UT1 and UTC. The value of the UT1UTC keyword shall be the difference between UT1 and UTC in seconds on the reference date for the array.

DATUTC. The value of the DATUTC keyword shall be the difference between the time system used in the data set and UTC in seconds on the reference date for the array. If the time system is IAT, this always has an integral value and is the number of accumulated leap seconds on that date. If it is UTC, the value is 0.

Time system. The TIMSYS keyword shall specify the time system used for the array. It shall either have the value 'IAT', denoting international atomic time, or the value 'UTC', denoting coordinated universal time. This indicates whether the zero hour for the TIME parameter in the UV_DATA table is midnight IAT or midnight UTC.

Array name. The value of the ARRNAM keyword shall be a name for the array that may be used in reports presented to human readers. Array names need not be unique and should not require more than 8 characters. Note, however, that software makes use of the array name making inventiveness in this parameter unwise.

Handedness. The value of the XYZHAND keyword should indicate whether the coordinate system used for antenna coordinates is right- or left-handed. The sign of the y coordinate reverses between RIGHT' and 'LEFT' and *AIPS* insists on converting antenna files to right-handed. In a right-handed system, an antenna in the Eastern United States has $x > 0$ and $y < 0$, one in the Western United States has $x < 0$ and $y < 0$, one in Germany has $x > 0$ and $y > 0$, and one in Australia has $x < 0$ and $y > 0$.

Coordinate frame. The value of the FRAME keyword shall be a string that identifies the coordinate system used for antenna coordinates. At present, only one value of the FRAME keyword has been defined ('ITRF'), although '????' is widely used to reflect ignorance.

Orbital parameters. If the antenna file defines parameters for one or more antennas, them the number of parameters used to describe the orbit shall be given by the NUMORB keyword.

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the antenna file, this controls the dimension of the polarization calibration value column.

Number of polarization calibration constants. The ANTENNA table may carry information about the polarization characteristics of the feeds, on an IF basis, if this is known. If information about the polarization characteristics of the feeds is contained in the table, then the NOPCAL keyword shall have the value 2. If no information about the polarization characteristics is contained in the table, then the NOPCAL keyword shall have the value 0.

Polarization parametrization. If the table contains information about the polarization characteristics of the feeds, then the feed parametrization that is used shall be indicated by the value of the POLTYPE keyword, as given in Table 9.

TABLE 9: Values for the POLTYPE keyword

Value	Model
'APPROX'	Linear approximation for circular feeds
'X-Y LIN'	Linear approximation for linear feeds
'ORI-ELP'	Orientation and ellipticity
'VLBI'	VLBI solution form

Each row in the table gives the parameters for one antenna in the current subarray and frequency setup. Each of the columns listed in Table 10 shall be present. The order of the columns does not matter.

TABLE 10: Mandatory and optional columns for the AIPS AN table

Title	Type	Units	Description
ANNAME	8A		Antenna name
STABXYZ	3D	meters	Antenna station coordinates (x, y, z)
ORBPARM	D(n_{orb})		Orbital parameters
NOSTA	1J		Antenna number
MNTSTA	1J		Mount type
STAXOF	1E	meters	Axis offset
POLTYA	1A		'R', 'L', feed A
POLAA	1E	degrees	Position angle feed A
POLCALA	E(n_{pcal},n_{IF})		Calibration parameters feed A
POLTYB	1A		'R', 'L', polarization 2
POLAB	1E	degrees	Position angle feed B
POLCALB	E(n_{pcal},n_{IF})		Calibration parameters feed B
DIAMETER	1E	meters	Antenna diameter
BEAMFWHM	E(n_{IF})	degrees/m	Antenna beam FWHM

Antenna name. The antenna name shall be a non-blank character string that may be used to identify the antenna for a human user. The special name 'OUT' shall indicate that the specified antenna number was not used in the data set.

Station coordinates. The STABXYZ array shall give the coordinate vector (element 1 is the x coordinate, element 2 is the y coordinate, and element 3 is the z coordinate) of the antenna relative to the array center defined in the header, provided that the antenna is not an orbiting antenna. The coordinate system used for the antenna coordinates is indicated by the FRAME keyword in the header. In the absence of FRAME information, the coordinate system longitudes are assumed to be based in Greenwich, England if the array center is all zero and to be based at the array center if it is not zero. Note that this means that the x and y coordinates of the station are rotated wrt the coordinate system used to express the array center and so cannot be added to them in a simple fashion.

Orbital parameters. If the antenna is an orbiting antenna and orbital information is available, the ORBPARM array will contain the orbital parameters for the antenna as shown in Table 11. The orbital elements shall be those for 0 hours on the reference date for the array in the time system used for the array. The reference frame for the orbital parameters shall be the same as that used for u, v, w coordinates in the uv data.

TABLE 11: Contents of the ORBPARM array

Index	Parameter	Units
1	Semi-major axis of orbit (a)	meters
2	Ellipticity of orbit (e)	
3	Inclination of the orbit to the celestial equator (i)	degrees
4	The right ascension of the ascending node (Ω)	degrees
5	The argument of the perigee (ω)	degrees
6	The mean anomaly (M)	degrees

The dimension of the ORBPARM array is given by the value of the NUMORB keyword. If this value is zero, then the ORBPARM column contains no values. If n_{orb} is 6, then all 6 orbital parameters shall be set to NaN (not a number) for all antennæ for which MNTSTA is not 2.

Antenna number. The NOSTA column shall contain a positive integer value that uniquely defines the antenna within the subarray. If the same antenna appears in more than one subarray, it need not have the same station number in each array. This is the antenna identification number that is used in other tables, including the visibility data.

Mount type. The MNTSTA column shall contain an integer value that encodes the mount type of the antenna. Codes 0 for alt-azimuth, 1 for equatorial, 2 for orbiting, 3 for X-Y, 4 for right-handed Naismith, and 5 for left-handed Naismith are defined. Aperture arrays, which are steered electronically rather than mechanically, are assigned code 6.

Axis offset. The axis offset is the position of the antenna phase reference point in the Yoke, relative to the antenna pedestal reference point. This is an antenna characteristic that should be unchanged when the antenna is moved to a new station. The X component of the offset is horizontal along the elevation axis and has no effect on interferometer phase. The Z component is vertical and approximately the nominal height of the elevation axis above ground for the antenna's mount. Small variations from the nominal value have the same phase effect as the Z component of position, so they can be ignored. The value of the `STAXOF` column gives the value of the Y component of the axis offset. That component is horizontal and perpendicular to the elevation axis. It produces an elevation-dependent interferometer phase term and, thus, has to be accurately calibrated.

Polarization types. The value in the `POLTYA` column shall be the feed polarization of feed A. This corresponds to polarization 1 in calibration tables. The value in the `POLTYB` column shall be the feed polarization of feed B (if any). The two feeds may be either circularly or linearly polarized. Mixtures of linear and circular polarizations are forbidden. If two orthogonal polarizations are used, it is strongly recommended that feed A (`POLTYA`) be `'R'` or `'X'` and feed B (`POLTYB`) be `'L'` or `'Y'`.

Feed orientations. The value of the `POLAA` column shall be the orientation of feed A, assumed independent of IF, given in degrees. Similarly, the `POLAB` column shall contain the feed orientation for feed B.

Polarization parameters. If the value of the `NOPCAL` keyword is 2, then the `POLCA` and `POLCB` columns shall contain 2 polarization parameters for each IF for feeds A and B, respectively. If the value of the `POLTYPE` keyword is `'APPROX'` or `'X-Y LIN'`, then the first parameter shall be the real part of the leakage term and the second shall be the imaginary part of the leakage term. If the value of the `POLTYPE` keyword is `'OTI-ELP'`, then the first parameter shall be the orientation and the second shall be the ellipticity and both shall be given in radians.

Antenna diameter. The optional `DIAMETER` column shall give the antenna physical diameter. This information may be used in calculations of sensitivity and shadowing.

Antenna beam. The optional column `BEAMFWHM` shall contain the full-width at half maximum of the (single-dish) beam of the antenna. It shall be expressed in degrees per meter wavelength and shall be assumed to scale with actual observing wavelength within the corresponding IF.

4.2 AIPS CD CalDevice table

The `AIPS CD` or CalDevice table contains the noise tube values for each antenna and IF. It is used with the `SY` table to apply gain corrections and determine expected noise in the taskS `TYAPL` and `SYSOL` in $\mathcal{AIPS}$. It is also used in various displays of the `SY` table data as system temperature. An example `AIPS CD` table header is shown in Appendix B.4.2 on page 85.

TABLE 12: Mandatory keywords for AIPS CD table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS CD'
NO_ANT	I	Maximum antenna number
NO_POL	I	Number polarizations (n_{feed})
NO_IF	I	Number IFs (n_{IF})
RDATE	D	Reference date

Number of antennas. The value of the `NO_ANT` keyword shall specify the maximum antenna number to occur in the `AIPS CD` table.

Number of polarizations. The value of the `NO_POL` keyword shall specify the number of distinct polarizations (feeds) in the data set (1 or 2). In the CalDevice file, this controls the number of Tcal columns.

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the CalDevice file, this controls the dimension of the Tcal columns.

Reference date. The value of the RDATE parameter will be the date for which the Tcal apply.

TABLE 13: Mandatory and optional columns for the AIPS CD table

Title	Type	Units	Description
ANTENNA NO.	1J		Antenna number
SUBARRAY	1J		Subarray
FREQ ID	1J		Frequency setup number
TCAL1	E(n_{IF})	Kelvin	Noise tube polarization 1
TCAL2	E(n_{IF})	Kelvin	Noise tube polarization 2
SOLCAL1	E(n_{IF})	Kelvin	Solar noise tube polarization 1
SOLCAL2	E(n_{IF})	Kelvin	Solar noise tube polarization 2

Antenna number. The ANTENNA NO. column shall contain a positive integer value that uniquely defines the antenna within the subarray. This is the antenna identification number that is used in other tables, including the visibility data. If the same antenna appears in more than one subarray, it need not have the same station number in each array.

Subarray number. The SUBARRAY column shall contain a positive integer value that uniquely identifies the subarray number to which the other data in the table row apply.

Frequency setup number. The FREQ ID column shall contain a positive integer that uniquely identifies the frequency setup to which the other data in the row apply.

Noise tube temperatures. The TCAL1 column shall contain n_{IF} values of the switched noise tubes for polarization 1 in degrees Kelvin. If the value of keyword NO_POL is 2, the TCAL2 column shall contain n_{IF} values of the switched noise tubes for polarization 2 in degrees Kelvin.

Solar noise tube temperatures. The SOLCAL1 column shall contain n_{IF} values of the switched noise tubes used during observations of the Sun for polarization 1 in degrees Kelvin. If the value of keyword NO_POL is 2, the SOLCAL2 column shall contain n_{IF} values of the switched noise tubes for Solar observations in polarization 2 in degrees Kelvin. The Solar columns are optional. A value of exactly 1.0 is taken to mean that that antenna/IF has no Solar noise tube.

4.3 AIPS CQ correlator frequency parameter table

The AIPS CQ conveys the frequency parameters used by the correlator. In the FITS-IDI context, the AIPS CQ table is built during the translations from IDI to *AIPS* using data from the MODEL_COMPS AND ANTENNA IDI tables. It is widely used by VLBI and conveys most importantly information about frequency averaging (if any) done after correlation. Delay errors cause loss of amplitude when frequency channels are averaged, *but only after correlation.* **The EVLA needs this information.** Tables 14 and 15 list the keywords and columns of the AIPS CQ table. An example AIPS CQ table header is shown in Appendix B.4.3 on page 86.

TABLE 14: Mandatory keywords for AIPS CQ table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS CQ'
NO_IF	I	Number IFs (n_{IF})
TABREV	I	CQ table revision number (1)

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the correlator frequency file, this controls the dimension of the all columns except subarray and frequency setup.

Frequency setup number. The `FRQSEL` column shall contain a positive integer that uniquely identifies the frequency setup to which the other data in the row apply.

Subarray number. The `SUBARRAY` column shall contain a positive integer value that uniquely identifies the subarray number to which the other data in the table row apply.

Size of FFT. The numerical size of the FFT per spectral window used to convert from time domain to frequency prior to cross-correlation shall be specified in the `FFT_SIZE` column.

TABLE 15: Mandatory columns for the AIPS CQ table

Title	Type	Units	Description
FRQSEL	1J		Frequency setup number
SUBARRAY	1J		Subarray
FFT_SIZE	J(n_{IF})		FFT size
NO_CHAN	J(n_{IF})		Number of spectral channels
SPEC_AVG	J(n_{IF})		Spectral averaging factor
EDGE_FRQ	D(n_{IF})	Hz	Edge frequency
CHAN_BW	D(n_{IF})	Hz	Channel bandwidth
TAPER_FN	8A(n_{IF})		weighting function
OVR_SAMP	J(n_{IF})		oversampling factor
ZERO_PAD	J(n_{IF})		zero padding
FILTER	J(n_{IF})		time filter
TIME_AVG	E(n_{IF})	seconds	accumulation time
NO_BITS	J(n_{IF})		quantization
FFT_OVLP	J(n_{IF})		FFT overlap factor

Number of spectral channels. The current number of spectral channels per spectral window shall be given in the `NO_CHAN` column. Note that this number is initially one half of the FFT size and may be reduced further by channel averaging. It should not be changed by windowing in the spectral domain.

Spectral averaging factor. The number of spectral channels averaged together after correlation is specified in the `SPEC_AVG` column. It is normally $Nfft/(2Nchan)$.

Frequency at spectral channel 1. The `EDGE_FRQ` column shall contain the frequencies at the reference spectral channel (which is usually channel 1) for each IF.

Frequency increment between channels. The `CHAN_BW` column specifies the current separation of spectral channels in Hz for each IF.

Weighting function. In 8 characters per IF, the `TAPER_FN` column shall specify the tapering function applied to the time domain prior to FFT. Two values, `'HANNING'` and `'UNIFORM'` are recognized. The form of the correction factor for segmentation loss in the presence of delay error depends on the type of function used.

Oversampling factor and zero padding. The oversampling and zero padding "factors" are given by the `OVR_SAMP` and `ZERO_PAD` columns; a value of 0 for these implies no oversampling and no zero padding.

Time filter. The `FILTER` columns is used to convey $256 \times (n-1) + m$, where n is the "correlation id number and m is the filter number to which the current row applies. These may be random parameters in the *uv* data; see Table 6, page 10.

Accumulation time. The integration time of the data represented by this row in the table shall be recorded in the `TIME_AVG` column. In general, this value will be the integration time given in the random parameters of the *uv* data and will not depend on IF.

Number of bits. The log base 2 of the number of digitizer levels for the antennas. In principle, this can carry with antenna, but `FITLD` takes that of antenna 1 as representative. The number of levels shall be 2 for Mk II and Mk III terminals and may be either 2 or 4 for VLBA terminals (depending on observing mode).

FFT overlap. The `FFT_OVPL` column is not well defined and should contain 0.

4.4 AIPS CT Calc table

The AIPS CT table contains the parameters of Calc, the software package that computes the delay model used by the correlator. It is not a standard FITS-IDI table, but said to be VLBA-specific. Since the EVLA also uses Calc, it should probably also populate this table. The position of the North Pole assumed at correlation is used to correct data for later, presumably better, estimates of that parameter in task CLCOR. The Polar X and Y also occur in the AIPS AN file (Section 4.1 on page 12), but with only one value at an unknown time. The VLBA, at least, uses interpolation over a 5-day interval to determine the best value to use in the correlation. An example AIPS CT table header is shown in Appendix B.4.4 on page 87.

TABLE 16: Mandatory and optional keywords for AIPS CT table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS CT'
OBSCODE	A	Observer project code
RDATE	D	Reference date
NO_STKD	I	Number of polarization products
STK_1	I	Value of first polarization product
NO_BAND	I	Number IFs (n_{IF})
NO_CHAN	I	Number spectral channels n_{chan})
REF_FREQ	E	Reference frequency (Hz)
CHAN_BW	E	Spectral channel separation (Hz)
REF_PIXL	E	Spectral channel centered on reference frequency
TABREV	I	CT table revision number (3)
C_SRVR	A	Computer serving Calc to correlator
C_VERSN	A	Calc version used
A_VERSN	A	Calc subpackage version '2.2'
I_VERSN	A	Calc subpackage version '0.0'
E_VERSN	A	Calc subpackage version '9.1' same as C_VERSN
ACCELGRV	E	Acceleration of gravity at the Earth's surface (9.78031846ms^{-2}
E-FLAT	E	Earth's flattening factor (0.00335281)
EARTHRAD	E	Earth's Equatorial Radius (6378137m)
MMSEMS	E	Ratio of mass of Moon to mass of Earth (0.01230002)
EPHEPOC	I	Coordinate *equinox* (2000)
ETIDELAG	E	Earth tides: lag angle (0.0 radians)
GAUSS	E	Gaussian gravitational constant (0.01720209895)
GMMOON	E	Lunar-centric gravitational constant (4.90279750E12 m^3s^{-2})
GMSUN	E	Heliocentric gravitational constant (1.32712438E20 m^3s^{-2})
LOVE_H	E	Earth tides: global Love Number H, IERS value (0.60967)
LOVE_L	E	Earth tides: global Love Number H, IERS value (0.0852)
PRE_DATA	E	General precession in longitude at standard equinox J2000 (5029.0966 arcsec per Julian century)
REL_DATA	E	?? (1.0)
TIDALUT1	I	?? (0.0)
TSECAU	E	Light time for one A.U. (499.004782 sec)
U-GRV-CN	E	Constant of gravitation (6.672E-11 $m^3kg^{-1}s^{-2}$)
VLIGHT	E	speed of light (299792458 ms^{-1})

Observation code. The Observer project code assigned to the experiment is given as the value of the OBSCODE keyword.

Reference date. The value of the RDATE parameter will be the date for which the Tcal apply.

Polarization data. The number of polarization products in the data set, n_{stokes}, is given by the value of the NO_STKD keyword and the Stokes value of the first product is indicated by the value of the STK_1 keyword.

Number of spectral windows and channels. The value of the NO_BAND keyword shall specify the number of spectral windows (IFs) and the value of the keyword NO_CHAN shall specify the number of spectral channels in the data set.

Frequency data. The sky observing frequency in Hz at the center of the reference pixel is specified by the value of the REF_FREQ keyword and the reference pixel itself is given by the value of the REF_PIXL keyword. The increment in frequency between spectral channel is given by the value of the CHAN_BW keyword.

Calc software used. The computer used to serve Calc to the correlator is indicated by the value of the C_SRVR keyword. The particular version numbers of the full and specific subsections of the Calcsoftware package are indicated by the values of the C_VERSN, A_VERSN, I_VERSN, and E_VERSN keywords, respectively. The meanings of the A, I, and E have been lost.

Calc model parameters. The remaining optional keywords in the AIPS CT table describe the astrometric model parameters provided to Calc. Their definitions are shown in Table 16 along with the values used for them by the DiFX software correlator.

TABLE 17: Mandatory columns for the AIPS CT table

Title	Type	Units	Description
TIME	1D	days	time of center of interval
UT1-UTC	1D	seconds	difference UT1 time - UTC time
IAT-UTC	1D	seconds	difference IAT time - UTC time
A1-IAT	1D	seconds	difference A1 time - UTC time
UT1 TYPE	1A		E extrapolated, P preliminary, F final
WOBXY	2D	arcseconds	X,Y polar offsets
WOB TYPE	1A		E extrapolated, P preliminary, F final
DPSI	1D	radians	nutation in longitude
DDPSI	1D	radians/sec	derivative of DPSI
DEPS	1D	radians	nutation in obliquity
DDEPS	1D	radians/sec	derivative DEPS
TIME INT	2D	days	range in time to which row applies

Time. The time in days since the reference date for the center of the interval is given in the TIME column.

Time systems. The difference between UT1 and UTC times is given in the UT1-UTC column, the difference between IAT (actually tai) abd UTC is given in the IAT-UTC column, and the difference between the A1 (whatever that is) and UTC times is given in the A1-UTC column. The UT1 TYPE column gives a single character indicating the origin of the UT1-UTC value — extrapolated, preliminary, final, or 'X' for unknown.

Position of the North Pole. The offset of the Earth's rotational North pole from nominal at the stated time is given in arc seconds in the two values per row WOBXY column. The origin (quality) of these values is indicated by the WOB TYPE column.

Nutation. The nutation in longitude and its derivative are given in the DPSI and DDPSI columns, respectively. The nutation in obliquity and its derivative are given in the DEPS and DDEPS columns, respectively. Since the values used by Calc are hard to extract, the DiFX correlator fills these columns with zeros.

Time interval. The start and stop times of the observation period for which the row applies are given by the TIME INT columns. If this column is missing or filled with zeros, CLCOR attempts to determine the appropriate values from the history file FITLD JOBSTART and FITLD JOBSTOP cards. FITLD updates the AIPS CT table when reading FITS-IDI files to fill these columns correctly.

4.5 AIPS FG flag table

The AIPS FG table contains commands to suppress (flag, delete) portions of the *uv* data set. It should begin with information from the on-line system for such things as "antenna not on source", "subreflector out of position", and the like. It could also have information from the correlator for known failures in correlation. The AIPS FG table, or multiple versions build from the first one, will acquire additional commands from auto-flagging tasks such as RFLAG. interactive flagging tasks such as TVFLG, or directly from the user with UVFLG. It would be possible to devise an improved online-flags table (Section 4.13, page 31) which then could be translated to an initial flag table inside AIPS with an enhanced version of the task OFLAG. An example AIPS FG table header is shown in Appendix B.4.5 on page 89.

There are no required keywords in the AIPS FG table. The columns required for the table are shown in Table 18.

TABLE 18: Mandatory columns for the AIPS FG table

Title	Type	Units	Description
SOURCE	1J		Source ID number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup number
ANTS	2J		antennas
TIME RANGE	2E	days	time range
IFS	2J		IF range
CHANS	2J		Spectral channel range
PFLAGS	4L		Stokes selection
REASON	24A		Reason for flag

Source. If the file contains observations of more than one source, then the identification number of the source to be flagged shall be given as the value in the SOURCE column. A value ≤ 0 means to flag data from all sources regardless of identification number.

Array and frequency setup numbers. The subarray and frequency selection numbers of the data to be flagged are specified in the SUBARRAY and FREQ ID columns, respectively. A value ≤ 0 means to flag data from all subarrays and/or frequency setups.

Antennas. The numbers assigned to the two antennas of the baseline pair to be flagged are given by the values in the ANTS column. If one number of the pair is zero, then all baselines involving the other antenna are to be flagged. If both numbers are zero, then all baselines are to be flagged.

Times. The beginning and ending times of the time range to be flagged are specified in the TIME RANGE column in days from the reference date. There is no default value, all numbers are taken literally.

Spectral windows. The beginning and ending spectral window numbers (IFs) to be flagged are found in the IFS column. A value of zero as the beginning IF number is interpreted as 1 and a value of 0 in the second IF number is interpreted as the maximum IF number (n_{IF}).

Spectral channels. The beginning and ending spectral channel numbers to be flagged in each spectral window are found in the CHANS column. A value of zero as the beginning channel number number is interpreted as 1 and a value of 0 in the second channel number is interpreted as the maximum number (n_{chan}).

Polarization selection. The PFLAGS column shall contain four logical values to specify which of the Stokes parameters are to be flagged.

Reason. The REASON column contains an arbitrary string of 24 characters to provide an explanation of the flag to the user. Typical values might be "antenna off source" and "TVFLG 14JUN11 14:53" for on-line and interactive reasons. UVFLG allows the user to specify the reason and to select flags based on reason.

4.6 AIPS FO frequency offset table

The AIPS FO table is a recently invented table to convey information about frequency changes occurring during the observation. Frequently spectral-line observations are made while keeping the "velocity" of a particular spectral channel constant. To do this as the Earth rotates about its axis and about the Sun, the observing frequency must change. Some tasks such as those that recompute the projected baselines and those that shift the spectra to do the "Doppler tracking" should care in detail exactly what the observing frequency was for each sample. *AIPS* hid this information, obtained from the on-line data, in a column in the calibration table. That is inconvenient for other software packages that do not support AIPS CL tables (Section 5.5 on page 47) and fails completely for single-source files which have no CL table. The AIPS FO table replaces the usage of the "hidden" column of the CL table and is maintained with single- as well as multi-source data sets. An example AIPS FO table header is shown in Appendix B.4.6 on page 89.

TABLE 19: Mandatory and optional keywords for AIPS FO table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS FO'
NO_ANT	I	Maximum antenna number
NO_IF	I	Number IFs (n_{IF})
REVISION	I	File format revision code (1)

Number of antennas. The value of the NO_ANT keyword shall specify the maximum antenna number to occur in the AIPS FO table.

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the frequency offset table, this controls the dimension of the Doppler offset column.

TABLE 20: Mandatory columns for the AIPS FO table

Title	Type	Units	Description
TIME	1D	days	time of center of interval
TIME INTERVAL	1E	days	length of time interval
SOURCE ID	1J		Source ID number
ANTENNA NO.	1J		Antenna number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup number
DOPOFF	E(n_{IF})	Hz	Doppler offset

Time. The time in days since the reference date for the center of the interval represented by the table row is given in the TIME column. The length of that interval is given in the TIME INTERVAL column.

Source identification number. If the file contains observations of more than one source, then the identification number of the source being observed shall be given as the value of the SOURCE ID column. A value ≤ 0 is taken to apply to all sources.

Antenna number. The ANTENNA NO. column shall contain a positive integer value that uniquely defines the antenna within the subarray. This is the antenna identification number that is used in other tables, including the visibility data. If the same antenna appears in more than one subarray, it need not have the same station number in each array.

Subarray number. The SUBARRAY column shall contain a positive integer value that uniquely identifies the subarray number to which the other data in the table row apply.

Frequency setup number. The FREQ ID column shall contain a positive integer that uniquely identifies the frequency setup to which the other data in the row apply.

Doppler offset. The DOPOFF column conveys n_{IF} values giving the actual observed frequency minus the time-independent frequency for each IF. The time-independent frequencies are described for each IF and frequency setup number in the AIPS FQ table (Section 4.7 on page 23) plus the frequency offsets described for each source in the AIPS SU (Section 4.17 on page 35) table plus the reference frequency given in the uv-data header.

4.7 AIPS FQ frequency table

The AIPS FQ or frequency table is an essential table unless there is only one spectral window and one frequency setup. It conveys the frequency offsets to be added to the reference frequency as a function of spectral window and frequency setup. The actual observing frequency requires the addition of source-dependent frequency offsets from the AIPS SU table (Section 4.17 on page 35) and time-dependent frequency offsets from the AIPS FO table (Section 4.6 on page 22). An example AIPS FQ table header is shown in Appendix B.4.7 on page 90.

TABLE 21: Mandatory and optional keywords for AIPS FQ table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS FQ'
NO_IF	I	Number IFs (n_{IF})

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the frequency table, this controls the dimension of all columns except the frequency setup number.

TABLE 22: Mandatory and optional columns for the AIPS FQ table

Title	Type	Units	Description
FRQSEL	1J		frequency setup ID number
IF FREQ	D(n_{IF})	Hz	Frequency offset
CH WIDTH	E(n_{IF})	Hz	Spectral channel separation
TOTAL BANDWIDTH	E(n_{IF})	Hz	Total width of spectral window
SIDEBAND	J(n_{IF})		Sideband
BANDCODE	8A(n_{IF})		Band/receiver code

Frequency setup number. The FRQSEL column shall contain a positive integer that uniquely identifies the frequency setup to which the other data in the row apply.

Frequency offset. The offset of the particular frequency setup and spectral window in Hz is given by the value contained in the IF FREQ column.

Frequency widths. The separation of spectral channels in Hz is given by the values contained in the CH WIDTH column as a function of IF. The total width of the spectral windows is given by the values contained in the TOTAL BANDWIDTH column. Thus the number of spectral channels is $T_{bw}(IFno)/|C_{width}(IFno)|$ in IF $IFno$.

Sideband. A value of -1 is used to indicate lower sideband and $+1$ upper sideband. Normally, the sign of SIDEBAND equals the sign of CH WIDTH. The actual increment in frequency when channel number increases by one is the value contained in the CH WIDTH column times the value contained in the SIDEBAND column. It is the practise in AIPS to reverse the order of spectral channels from lower sideband data and to then call the data upper sideband.

Band code. The optional column BANDCODE contains one 8-character, blank-filled string for each spectral window. The string identifies the receiver (and feed) used to observe the data of the particular frequency setup and spectral window. The form of the string is undefined except for the EVLA where it is either x or EVLA_x, where x is one of '4', 'P', 'L', 'S', 'C', 'X', 'U', 'K', 'A', or 'Q'.

4.8 AIPS GC gain curve table

The AIPS GC table may be used to perform the entire time-independent portion of the amplitude calibration. It conveys the relative gains of each antenna as a function of pointing position as well as the point-source sensitivity. The product of these, evaluated at the observing frequency at pointing position, is the amplitude gain. This method has been widely used with VLBI data through *AIPS* task APCAL. For VLA data, *AIPS* has preferred to use tabulated antenna gain tables in creating the initial calibration (AIPS CL) to correct for antenna relative gains. Additional details are given in the FITS IDI Convention Memo (Greisen 2011 [8]).

TABLE 23: Mandatory and optional keywords for AIPS GC table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS GC'
NO_BAND	I	Number IFs (n_{IF})
NO_POL	I	Number polarizations (n_{feed})
NO_TABS	I	Number of values in table (n_{tab})
TABREV	I	Table revision number (2)

Number of spectral windows. The value of the NO_BAND keyword shall specify the number of spectral windows (IFs) in the data set. In the gain curve table, this controls the dimension of all columns except the antenna, subarray, and frequency setup numbers.

Number of polarizations. The value of the NO_POL keyword shall specify the number of distinct polarizations (feeds) in the data set (1 or 2). In the gain curve file, this controls whether the columns named with a "2," those below the line, appear.

Number of table values. The value of the NO_TABS keyword shall specify the number of table values in the Y_VAL_n and GAI_n columns.

TABLE 24: Mandatory columns for the AIPS GC table

Title	Type	Units	Description
ANTENNA_NO	1J		antenna number
SUBARRAY	1J		subarray number
FREQ ID	1J		frequency setup ID number
TYPE_1	J(n_{IF})		Gain curve type, feed A
NTERM_1	J(n_{IF})		Number of terms or entries, feed A
X_TYP_1	J(n_{IF})		Type of X coordinate, feed A
Y_TYP_1	J(n_{IF})		Type of Y coordinate, feed A
X_VAL_1	E(n_{IF})	degrees	Value of X coordinate, feed A
Y_VAL_1	E(n_{tab},n_{IF})	degrees	Value of Y coordinate, feed A
GAIN_1	E(n_{tab},n_{IF})		Gain table, feed A
SENS_1	E(n_{IF})	Kelvin/Jy	Sensitivity table, feed A
TYPE_2	J(n_{IF})		Gain curve type, feed B
NTERM_2	J(n_{IF})		Number of terms or entries, feed B
X_TYP_2	J(n_{IF})		Type of X coordinate, feed B
Y_TYP_2	J(n_{IF})		Type of Y coordinate, feed B
X_VAL_2	E(n_{IF})	degrees	Value of X coordinate, feed B
Y_VAL_2	E(n_{tab},n_{IF})	degrees	Value of Y coordinate, feed B
GAIN_2	E(n_{tab},n_{IF})		Gain table, feed B
SENS_2	E(n_{IF})	Kelvin/Jy	Sensitivity table, feed B

Antenna number. The ANTENNA_NO column shall contain a positive integer value that uniquely defines the antenna within the subarray. This is the antenna identification number that is used in other tables, including the visibility data. If the same antenna appears in more than one subarray, it need not have the same station number in each array.

Subarray number. The SUBARRAY column shall contain a positive integer value that uniquely identifies the subarray number to which the other data in the table row apply.

Frequency setup number. The FREQ ID column shall contain a positive integer that uniquely identifies the frequency setup to which the other data in the row apply.

Gain curve type. The value of the TYPE_n columns shall specify the type of gain curve as 1 tabulated values, 2 polynomial coefficients, or 3 spherical harmonic coefficients.

Number of terms. The value of the NTERM_n columns shall specify the number of tabulated values or the number of coefficients in each gain curve.

Axis type code. The values in the X_TYP_n and Y_TYP_n columns shall specify the meaning of X and Y axis values as 0 none, 1 elevation, 2 zenith angle, 3 hour angle, 4 declination, and 5 90-declination. the position of a datum to be calibrated is computed from its observer-relative coordinates of the specified type.

Axis values. The values of the X_VAL_Un column gives the X coordinate value, one per IF, appropriate to the row. The values of the Y_VAL_Un column gives the Y coordinate value, one per term per IF, appropriate to the row. The X coordinate is used with to select the nearest gain values to the X position of the datum being calibrated. The Y coordinate is only relevant with tabulated values (gain type 1) and is used to interpolate the gain to be used for the particular datum.

Gain values. The values of the GAIN_n column contain the corresponding relative gain values (if type 1) or relative gain coefficients (types 2 and 3). For polynomial gain type, the relative gain at the datum having position Y on the Y axis is

$$g = \sum_{i=1}^{n_{term}} Y^{i-1} G(i). \tag{1}$$

For spherical harmonic gain type, a much more complicated computation is performed using the X and Y coordinates of the datum; see AIPS task APCAL for details.

Sensitivity The values of the SENS_n column contain the point source sensitivity in degrees Kelvin per Jansky for each IF. The net gain applied to a datum is the product of the relative gain determined using the other columns times the point source sensitivity.

4.9 AIPS IM interferometer model table

The AIPS IM table contains information about the interferometer models used by the correlator. In AIPS it is used to fill in the delay polynomial and dispersion columns in the initial version of the calibration table (Section 5.5 on page 47). An example AIPS IM table header is shown in Appendix B.4.8 on page 91.

TABLE 25: Mandatory keywords for AIPS IM table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS IM'
RDATE	D	Reference date
OBSCODE	A	Observation code
NO_STKD	I	Number of Stokes products (n_{stokes})
STK_1	I	First Stokes value
NO_BAND	I	Number of IFs (n_{IF})
NO_CHAN	I	Number of spectral channels (n_{chan})
REF_FREQ	E	Reference frequency (Hz)
CHAN_BW	E	Separation of spectral channels (Hz)
REF_PIXL	E	Reference spectral channel (pixels)
NO_POL	I	Number of polarizations
NPOLY	I	Number of polynomial terms n_{poly}
REVISION	I	Revision number of correlator software
TABREV	I	IM table format revision (2)

FITS-IDI required keywords. All FITS-IDI tables are required to contain the familiar keywords for reference date, observation code, Stokes values, number of IFs and spectral channels, and frequency description.

Number of polarizations. The `AIPS IM` table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the `NO_POL` keyword.

Number of polynomial terms. Delays and rates are given as polynomials with n_{poly} terms as specified by the value of the `NPOLY` keyword. This shall be a positive integer.

Each row of the table shall give the model information applicable to one antenna over a range of time. Each of the columns listed in Table 26 above the horizontal line must be present. The columns for the second polarization, listed below the horizontal line, must appear but only if the value of the `NO_POL` keyword is two. Polarization 1 corresponds to feed A in the `ANTENNA` table and polarization 2 to feed B. The order of the columns does not matter.

TABLE 26: Mandatory columns for the `AIPS IM` table

Title	Type	Units	Description
TIME	1D	days	Starting time of interval
TIME_INTERVAL	1E	days	Duration of interval
SOURCE_ID	1J		Source ID number
ANTENNA_NO	1J		Antenna number
ARRAY	1J		Array number
FREQID	1J		Frequency setup number
I.FAR.ROT	1E	rad m^{-2}	Ionospheric Faraday rotation
FREQ.VAR	E(n_{IF})	Hz	Time variable frequency offsets
PDELAY_1	D(n_{poly},n_{IF})	turns	Phase delay polynomials for polarization 1
GDELAY_1	D(n_{poly},n_{IF})	seconds	Group delay polynomials for polarization 1
PRATE_1	D(n_{poly},n_{IF})	Hz	Phase delay rate polynomials for polarization 1
GRATE_1	D(n_{poly},n_{IF})	sec/sec	Group delay rate polynomials for polarization 1
DISP_1	1E	sec m^{-2}	Dispersive delay for polarization 1
DDISP_1	1E	sec m^{-2}/sec	Rate of change of dispersive delay for polarization 1
PDELAY_2	D(n_{poly},n_{IF})	turns	Phase delay polynomials for polarization 2
GDELAY_2	D(n_{poly},n_{IF})	seconds	Group delay polynomials for polarization 2
PRATE_2	D(n_{poly},n_{IF})	Hz	Phase delay rate polynomials for polarization 2
GRATE_2	D(n_{poly},n_{IF})	sec/sec	Group delay rate polynomials for polarization 2
DISP_2	1E	sec m^{-2}	Dispersive delay for polarization 2
DDISP_2	1E	sec m^{-2}/sec	Rate of change of dispersive delay for polarization 2

Time covered by the row. The `TIME` column shall contain the earliest time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array. This is also the zero time for the delay and rate polynomials. The `TIME_INTERVAL` column shall contain the number of days for which the model described by the row remains valid. Note that the `AIPS IM` table differs from other *AIPS* tables in that the value in the `TIME` column is the beginning of the interval covered and not the center of the interval.

Source identification number. The `SOURCE_ID` column shall contain the source identification number of the source for which the model is valid.

Antenna and subarray numbers. The `ANTENNA_NO` column shall contain the antenna identification number of the antenna to which the model applies. The `ARRAY` column shall contain the number of the subarray to which the antenna belongs.

Frequency setup number. The `FREQID` column shall contain the frequency setup number of the frequency setup for which the model applies.

Ionospheric Faraday rotation. The I.FAR.ROT column shall contain the value of the ionospheric Faraday rotation applied at the correlator. If no correction has been applied, then this field shall contain 0.0.

Time variable frequency offsets. The FREQ.VAR column shall contain an array of time-variable frequency offsets that were applied. There shall be one entry for every IF in the file and each entry shall contain the additional frequency offsets applied to the IF as a function of time.

Phase and group delay polynomials. The GDELAY_1 and GDELAY_2 columns shall contain polynomial terms for the group delays for each band in polarization 1 and 2, respectively. The group delay is calculated from these according to Eq. 2, where Δt is the number of seconds that have elapsed since the beginning of the interval covered by the model and p_i is the polynomial term with index i for the current band.

$$\tau_g = \sum_{i=1}^{n_{poly}} p_i \cdot (\Delta t)^{i-1} \ . \tag{2}$$

The PDELAY_1 and PDELAY_2 columns shall contain the polynomial terms for the phase delay evaluated at the reference frequency for each band in the same format.

Phase and group delay rates. The GRATE_1 and GRATE_2 columns shall contain polynomial terms for the group delay rates (*i.e.*, the time derivatives of the group delays) for each IF in polarizations 1 and 2, respectively. Similarly, the PRATE_1 and PRATE_2 columns shall contain the polynomial terms for the phase delay rates. The same conventions are used as for the group delay terms. Note that the rate terms may be expected to be approximately equal to the delay terms but shifted by one position, but that exact equivalence is not required. This allows for correlators such as the VLBA which model delay and rate separately.

Dispersive delays. The DISP_1 and DISP_2 columns shall contain the components of the group delays for polarization 1 and 2 that scale with the square of the wavelength (*e.g.*, ionospheric delay). These shall be specified by giving the delays in seconds per meter squared. The DDISP_1 and DDISP_2 columns give the time derivatives of the dispersive delays in DISP_1 and DISP_2.

4.10 AIPS MC correlator model components table

The AIPS MC table is used to convey the parameters of the spectral sampling and of the various delay corrections applied to the data during the correlation. It was initially a VLBA-specific table, but has received wider use. In AIPS it is used to fill in the clock and atmosphere correction columns in the initial version of the calibration table (Section 5.5 on page 47). An example AIPS MC table header is shown in Appendix B.4.9 on page 92.

The mandatory keywords of the AIPS MC table are shown in Table 27 on the next page.

FITS-IDI required keywords. All FITS-IDI tables are required to contain the familiar keywords for reference date, observation code, Stokes values, number of IFs and spectral channels, and frequency description.

Number of polarizations. The AIPS MC table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the NO_POL keyword.

Size of FFT. The numerical size of the FFT per spectral window used to convert from time domain to frequency prior to cross-correlation shall be specified in the FFT_SIZE keyword.

Oversampling factor and zero padding. The oversampling and zero padding "factors" are given by the OVERSAMP and ZERO_PAD keywords; a value of 0 for these implies no oversampling and no zero padding.

Weighting function. The TAPER_FN keyword shall specify the tapering function applied to the time domain prior to FFT. Two values, 'HANNING' and 'UNIFORM' are recognized.

Interval between rows in table. The DELTAT keyword will specify the expected interval between rows in the table in days. The data are given for an antenna and one or more sources every DELTAT days and apply until the next recorded time.

TABLE 27: Mandatory keywords for AIPS MC table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS MC'
RDATE	D	Reference date
OBSCODE	A	Observation code
NO_STKD	I	Number of Stokes products (n_{stokes})
STK_1	I	First Stokes value
NO_BAND	I	Number of IFs (n_{IF})
NO_CHAN	I	Number of spectral channels (n_{chan})
REF_FREQ	E	Reference frequency (Hz)
CHAN_BW	E	Separation of spectral channels (Hz)
REF_PIXL	E	Reference spectral channel (pixels)
NO_POL	I	Number of polarizations
FFT_SIZE	I	FFT size
OVERSAMP	I	Oversampling factor
ZERO_PAD	I	Zero padding factor
TAPER_FN	8A	Time smoothing function prior to the FFT
DELTAT	E	Time interval in the table (days)
TABREV	I	IM table format revision (1)

Each row contains the various delays and frequency offsets for one antenna, source, and frequency setup for a limited range of time. Each row shall contain the columns shown in Table 28 above the horizontal line. Columns for the second polarization, listed below the horizontal line, must also appear but only if the value of the NO_POL keyword is two. The columns may be written in any order.

TABLE 28: Mandatory columns for the AIPS MC table

Title	Type	Units	Description
TIME	1D	days	Start time of interval
SOURCE_ID	1J		Source ID number
ANTENNA_NO	1J		Antenna number
ARRAY	1J		Array number
FREQID	1J		Frequency setup number
ATMOS	1D	sec	Atmospheric delay
DATMOS	1D	sec/sec	Time derivative of atmospheric delay
GDELAY	1D	sec	Group delay
GRATE	1D	sec/sec	Rate of change of group delay
CLOCK_1	1D	sec	"Clock" epoch error
DCLOCK_1	1D	sec/sec	Time derivative of clock error
LO_OFFSET_1	E(n_{IF})	Hz	LO offset
DLO_OFFSET_1	E(n_{IF})	Hz/sec	Time derivative of LO offset
DISP_1	1E	sec m^{-2}	Dispersive delay
DDISP_1	1E	sec m^{-2}/sec	Time derivative of dispersive delay
CLOCK_2	1D	sec	"Clock" epoch error
DCLOCK_2	1D	sec/sec	Time derivative of clock error
LO_OFFSET_2	E(n_{IF})	Hz	LO offset
DLO_OFFSET_2	E(n_{IF})	Hz/sec	Time derivative of LO offset
DISP_2	1E	sec m^{-2}	Dispersive delay
DDISP_2	1E	sec m^{-2}/sec	Time derivative of dispersive delay

Time covered by the row. The TIME column shall contain the number of days that have elapsed between 0 hours on the reference date for the current array and the *beginning* of the time period covered by the current row.

Source identification number. The SOURCE_ID column shall contain the source identification number of the source to which the current row applies. A value of zero in the SOURCE_ID column shall be understood to apply to all source identification numbers.

Antenna identification. The ANTENNA_NO column shall contain the antenna identification number and the ARRAY column shall contain the array number of the antenna to which the current row applies.

Frequency information. The FREQID column shall contain the frequency setup number of the frequency setup to which the current row applies.

Atmospheric delay. The ATMOS and DATMOS columns shall contain the atmospheric group phase delay and rate of change of that delay, respectively, applied to the data by the correlator software.

Group delay. The GDELAY and GRATE columns shall contain the group delay calculated by the CALC software and the rate of change of that delay, respectively, applied to the data by the correlator software.

Clock error. The CLOCK_1 and DCLOCK_1 columns shall contain the electronic, clock-like delay and the rate of change of that delay, respectively, applied to the data of polarization 1 by the correlator software. If the value of the NO_POL keyword is two, then the CLOCK_2 and DCLOCK_2 columns shall contain the electronic, clock-like delay and the rate of change of that delay, respectively, applied to the data of polarization 2 by the correlator software.

LO offset. The LO_OFFSET_1 and DLO_OFFSET_1 columns shall contain the station-dependent local oscillator offset and rate of change of that offset, respectively, applied to the data of polarization 1 by the correlator software. If the value of the NO_POL keyword is two, then the LO_OFFSET_2 and DLO_OFFSET_2 columns shall contain the station-dependent local oscillator offset and rate of change of that offset, respectively, applied to the data of polarization 2 by the correlator software.

Dispersive delays. The DISP_1 and DDISP_1 columns shall contain the component of the group delay that scales with the square of the wavelength (*e.g.*, ionospheric delay) and rate of change of that delay, respectively, applied to the data of polarization 1 by the correlator software. These shall be specified by giving the delays in seconds per meter squared. If the value of the NO_POL keyword is two, then the DISP_1 and DDISP_1 columns shall contain the component of the group delay that scales with the square of the wavelength (*e.g.*, ionospheric delay) and rate of change of that delay, respectively, applied to the data of polarization 2 by the correlator software.

4.11 AIPS NX index table

AIPS tasks prepare and maintain an AIPS NX table to reduce the time it takes to find desired visibilities in a *uv* data set. It changes, at least potentially, every time a data set is copied and so has little lasting value. It should however be written from the on-line system and correlator initially so that the "scan" structure of the observations may be known. *AIPS* tasks now create new index tables using the time breaks given in the input index table, thereby attempting to maintain that original scan structure. An example AIPS NX table header is shown in Appendix B.4.10 on page 94.

The AIPS NX table has no required keywords.

TABLE 29: Mandatory columns for the AIPS NX table

Title	Type	Units	Description
TIME	1D	days	Center time of interval
TIME INTERVAL	1E	days	Duration of interval
SOURCE ID	1J		Source number
SUBARRAY	1J		Subarray
FREQ ID	1J		Frequency setup ID number
START VIS	1J		First visibility number
END VIS	1J		End visibility number

Time. The time in days since the reference date for the center of the interval represented by the table row is given in the TIME column. The length of that interval is given in the TIME INTERVAL column.

Source identification number. If the file contains observations of more than one source, then the identification number of the source being observed shall be given as the value of the SOURCE ID column. A value ≤ 0 is taken to apply to all sources.

Subarray and frequency setup numbers. The SUBARRAY column shall contain the number of the subarray and the FREQ ID column shall contain the frequency setup number to which this table row applies.

Visibility record numbers. The START VIS and END VIS column shall the beginning visibility number and ending visibility number of the data described by the contents of the other columns. Visibilities are counted in order in the data set from one where a single visibility record is the equivalent of a random group as described in Section 3.1 on page 7.

4.12 AIPS OB spacecraft orbit table

The AIPS OB table is used to describe the elements of a spacecraft orbit. Since there are no spacecraft currently providing data to *AIPS*, this is of limited interest although it was used with the VSOP spacecraft in its day. It is nonetheless documented here for completeness. CLCOR uses this table, if present, when correcting for the relativistic time delay caused by the bending of light in the Sun's gravitational field and UVFIX uses it to compute projected baselines to the spacecraft.

There are no required keywords in the AIPS OB table other than TABREV which currently has the value 1.

TABLE 30: Mandatory columns for the AIPS OB table

Title	Type	Units	Description
ANTENNA_NO	1J		Antenna number
SUBARRAY	1J		Subarray number
TIME	1D	days	Time
ORBXYZ	3D	meters	Spacecraft position (x, y, z)
VELXYZ	3D	m/sec	Spacecraft velocity
SUN_ANGLE	3E	degrees	Sun angle
ECLIPSE	4E	days	Eclipse times
ORIENTATION	1E	degrees	Orientation

Antenna and subarray numbers. The ANTENNA_NO column shall contain the antenna identification number of the antenna to which the orbit applies. The SUBARRAY column shall contain the number of the subarray to which the antenna belongs.

Time covered by the row. The TIME column shall contain the number of days that have elapsed between 0 hours on the reference date for the current array and the moment in time to which the parameters in the row apply.

Antenna location. The ORBXYZ array shall give the coordinate vector (element 1 is the x coordinate, element 2 is the y coordinate, and element 3 is the z coordinate) of the antenna relative to the array center defined in the AIPS AN table header. The coordinate system used for the antenna coordinates is indicated by the FRAME keyword in the AIPS AN table header.

Antenna velocity. The VELXYZ array shall give the rate of change of the coordinate vector of the antenna relative to the array center defined in the AIPS AN table header.

Angles to the Sun. The angle between the direction to the Sun and to the source being observed in reported in the first of the three values of the SUN_ANGLE column.

Eclipse times. The time since entering Earth's shadow and the time since leaving Earth's shadow are reported in the first and second values of the ECLIPSE column in days.

Orientation. The geometric rotation of the orientation of the linearly polarized feed in degrees is reported in the `ORIENTATION` column. See AIPS task `OBTAB` and Kogan 1997 [18].

4.13 AIPS OF on-line flags table

The `AIPS OF` table contains information about all flags generated by the on-line system (and correlator), regardless of severity. Translation programs such as AIPS' `FILLM` for VLA data and OBIT's `BDFIn` for EVLA data select among these on-line flags to generate the initial `AIPS FG` table ((Section 4.5 on page 21). The on-line flags table allows for detailed display of the flagging information and for changing the flags that appear in the `AIPS FG` table which are applied to the data. This capability has not been widely used, but if the table were to be enhanced for EVLA use, would be an alternative to the expectation that an `AIPS FG` table be provided by the on-line system, correlator, and translation program combination.

AIPS task `FILLM` fills this table for the VLA, while task `OFLAG` will apply the online flags to the `AIPS FG` table and task `PRTOF` will interpret to the printer the contents of the online flags table.

TABLE 31: Mandatory keywords for AIPS OF table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS OF'
NO_IF	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
ANNAME	A	Array name
OBSCODE	A	Observation code
RDATE	D	Reference date
REVISION	I	OF table format revision (2)

Number of spectral windows. The value of the `NO_IF` keyword shall specify the number of spectral windows (IFs) in the data set. In the online flags table, this controls the dimension of the status columns.

Number of polarizations. The `AIPS OF` table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the `NO_POL` keyword.

Array name. The value of the `ANNAME` keyword shall be a name for the array that may be used in reports presented to human readers. Array names need not be unique and should not require more than 8 characters. Note, however, that software will need to use this to determine the meaning of the flag bit patterns.

Project code and date. The code used to identify the observation will be indicated by the value of the `OBSCODE` keyword and the reference date will be indicated by the `RDATE` keyword.

Each row shall contain the columns shown in Table 32 above the horizontal line. Columns for the second polarization, listed below the horizontal line, must also appear but only if the value of the `NO_POL` keyword is two. The columns may be written in any order.

TABLE 32: Mandatory columns for the AIPS OF table

Title	Type	Units	Description
TIME	1D	days	Time
SOURCE ID	1J		Source number
ANTENNA NO.	1J		Antenna number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
ANT FLAG	1J		Antenna-based flag bit pattern
STATUS 1	J(n_{IF})		Status level, polarization 1
STATUS 2	J(n_{IF})		Status level, polarization 2

Time covered by the row. The TIME column shall contain the number of days that have elapsed between 0 hours on the reference date for the current array and the moment in time to which the parameters in the row apply.

Source identification number. If the file contains observations of more than one source, then the identification number of the source being observed shall be given as the value of the SOURCE ID column. A value ≤ 0 is taken to apply to all sources.

Antenna and subarray numbers. The ANTENNA_NO column shall contain the antenna identification number of the antenna to which the orbit applies. The SUBARRAY column shall contain the number of the subarray to which the antenna belongs.

Frequency setup number. The FREQ ID column shall contain a positive integer that uniquely identifies the frequency setup to which the other data in the row apply.

Antenna flag bit pattern. The 32-bit integer value contained in the ANT FLAG column represents all possible online flag conditions, where a 1 in a given position means that that condition has been raised. Table 33 shows the interpretation of the bits for the VLA (not the EVLA).

TABLE 33: Interpretation of the antenna flags for the VLA

Bit	Value	Meaning
1	1	reference pointing requested, but not applied
2	2	antenna shadowed at source change time
3	4	antenna off source
4	8	first LO not locked
5	16	Tsys fluctuating
6	32	antenna flagged bad by operator
7	64	back-end total power out of range
8	128	back-end filters mis-set
9	256	L8 module not locked
10	512	L6 module not locked
11	1024	sub-reflector not in position
12	2048	source change in progress
13	4096	phase switching disabled
14	8192	round-trip phase correction disabled

Summary status. The STATUS 1 column conveys a summary "score" of the state of each IF in the first polarization. If there are two polarizations (NO_POL has value 2), the STATUS 2 column conveys a summary "score" of the state of each IF in the second polarization. For the VLA a score of 0 means no warning, scores of 1 through 3 a mild warning, and scores of 4 through 15 an increasingly severe warning. In practise, the only scores issued in recent years were 0 and 4.

4.14 AIPS OT over-the-top table

The AIPS OT table has a single, but significant, purpose. If the observation of a source is made "over the top" its elevation is actually π minus the elevation that the normal routines which compute elevation return. This means different elevations in fitting elevation effects in ELINT and in fitting antenna locations in LOCIT. The latter is particularly important, since the use of over-the-top observations in baseline fitting gives more reliable solutions than would be obtained if no over-the-top data were included. An example AIPS OT table header is shown in Appendix B.4.11 on page 94.

There are no required keywords in the AIPS OT table.

TABLE 34: Mandatory columns for the AIPS OT table

Title	Type	Units	Description
TIME	1D	days	Center time of interval
TIME INTERVAL	1E	days	Duration of interval
SOURCE ID	1J		Source number
ANTENNA	1J		Antenna number
OVER TOP	1L		True if over the top

Time. The time in days since the reference date for the center of the interval represented by the table row is given in the `TIME` column. The length of that interval is given in the `TIME INTERVAL` column.

Source identification number. If the file contains observations of more than one source, then the identification number of the source being observed shall be given as the value of the `SOURCE ID` column. A value ≤ 0 is taken to apply to all sources.

Antenna number. The `ANTENNA` column shall contain a positive integer value that uniquely defines the antenna within the subarray. This is the antenna identification number that is used in other tables, including the visibility data. If the same antenna appears in more than one subarray, it need not have the same station number in each array.

Over-the-top flag. The `OVER TOP` column contains a single logical value conveying whether the antenna was over the top or not in the specified interval.

4.15 AIPS PC phase cal table

Pulse-calibration tones are injected into the VLBA receivers and, in conjunction with the cable calibration system, can be used to measure and track instrumental variations in delay and phase between separate baseband converters. If the pulses are separated by 1 microsecond in time then the corresponding tones are separated by 1 MHz in frequency. The measured phases of the tones allow the connection of the phase across all IFs. The phase difference between two tones in the same IF allows the instrumental single-band delay for that IF to be estimated within a $N \times 2\pi$ ambiguity. The visibilities of a single calibrator scan may used to resolve this ambiguity.

TABLE 35: Mandatory keywords for AIPS PC table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS PC'
NO_BAND	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
NO_TONES	A	Number of pulse cal tones
TABREV	I	PC table format revision (1)

Number of spectral windows. The value of the `NO_BAND` keyword shall specify the number of spectral windows (IFs) in the data set. In the phase cal table, this controls the dimension of the data columns other than the cable calibration.

Number of polarizations. The `AIPS PC` table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the `NO_POL` keyword. In the phase cal table, the value of this keyword controls whether there is one set of data columns or two.

Number of tones. The value of the `NO_TONES` keyword shall specify the number of phase-cal tones per IF in the data. It controls the dimensions of the frequency, real, imaginary, and rate columns.

TABLE 36: Mandatory columns for the AIPS PC table

Title	Type	Units	Description
TIME	1D	days	Central time of interval
TIME_INTERVAL	1E	days	Length of interval
SOURCE_ID	1J		Source number
ANTENNA_ NO	1J		Antenna number
ARRAY	1J		Subarray number
FREQID	1J		Frequency setup ID number
CABLE_CAL	1D	seconds	Cable calibration delay
STATE 1	E(4,n_{IF})	percent	digitizer states, polarization 1
PC_FREQ_1	D(n_{tone},n_{IF})	Hz	phase tone frequencies, polarization 1
PC_REAL_1	E(n_{tone},n_{IF})		real part of phase cal, polarization 1
PC_IMAG_1	E(n_{tone},n_{IF})		imaginary part of phase cal, polarization 1
PC_RATE_1	E(n_{tone},n_{IF})	sec/sec	rate of change, polarization 1
STATE 2	E(4,n_{IF})	percent	digitizer states, polarization 2
PC_FREQ_2	D(n_{tone},n_{IF})	Hz	phase tone frequencies, polarization 2
PC_REAL_2	E(n_{tone},n_{IF})		real part of phase cal, polarization 2
PC_IMAG_2	E(n_{tone},n_{IF})		imaginary part of phase cal, polarization 2
PC_RATE_2	E(n_{tone},n_{IF})	sec/sec	rate of change, polarization 2

Time covered by the row. The TIME column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array. The TIME_INTERVAL column shall contain the number of days in the time interval represented by the row.

Source identification number. The SOURCE_ID column shall contain the source identification number of the source for which the data are valid.

Antenna and subarray numbers. The ANTENNA_NO column shall contain the antenna identification number of the antenna to which the data apply. The ARRAY column shall contain the number of the subarray to which the antenna belongs.

Frequency setup number. The FREQID column shall contain the frequency setup number of the frequency setup for which the data apply.

Cable calibration. The CABLE_CAL column shall contain the measured cable calibration delay in seconds.

Digitizer state. The STATE_1 column shall contain four values per IF represent the percentage of time that the digitizer spent in lowest, medium-low, medium-high, and highest states respectively for the first polarization. If the value of the NO_POL keyword is two, the STATE_2 column will contain similar values for the second polarization.

Phase tone frequencies. The PC_FREQ_1 column shall contain the phase tone frequencies for each tone and IF in polarization 1. If the value of the NO_POL keyword is two, the PC_FREQ_2 column will contain similar values for the second polarization.

Phase calibration measurement. The PC_REAL_1 column shall contain the real part of the phase calibration measurement for each tone and IF in polarization 1. The PC_IMAG_1 column shall contain the imaginary part of the phase calibration measurement for each tone and IF in polarization 1. The PC_RATE_1 column shall the rate of change of the phase-cal phase in Hz divided by the phase-cal frequency for each tone and IF in polarization 1. If the value of the NO_POL keyword is two, the PC_REAL_2, PC_IMAG_2, abd PC_RATE_2 columns will contain similar values for the second polarization.

4.16 AIPS PO position table

The AIPS PO table records the positions of moving sources (planets and the like) as a function of time. The source table (below) allows only one position for each source, making this extra table useful. This table is written by FILLM from VLA-format data, but no standard AIPS task makes use of it. An example AIPS PO table header is shown in Appendix B.4.12 on page 95.

TABLE 37: Mandatory keywords for AIPS PO table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS PO'
RDATE	D	Reference date
TABREV	I	PO table format revision (1)

Reference date. The value of the RDATE parameter will be the date on which the observations begin and from which time is measured in this data set.

TABLE 38: Mandatory columns for the AIPS PO table

Title	Type	Units	Description
TIME	1D	days	Time
SOURCE_ID	1J		Source number
SOURCE RA	1D	degrees	Source right ascension
SOURCE DEC	1D	degrees	Source declination
SOURCE DISTANCE	1D	au	Source distance

Time covered by the row. The TIME column shall contain the number of days that have elapsed between 0 hours on the reference date for the current array and the moment in time to which the parameters in the row apply.

Source identification number. If the file contains observations of more than one source, then the identification number of the source being observed shall be given as the value of the SOURCE ID column.

Source position. The apparent source position of date is given in the SOURCE RA and SOURCE DEC columns in degrees for the specified time. The source distance, if known, is given in the SOURCE DISTANCE column in astronomical units.

4.17 AIPS SU source table

For observations including more than one source, the AIPS SU table provides information on the names, calibration codes, fluxes, and coordinates of each source, The file is used extensively throughout AIPS and is required to be present if a SOURCE random parameter appears in the random-groups data set or a SOURCE column appears in the UV-tables data set. The file is normally not present when the SOURCE parameter (column) is not present. An example AIPS SU table header is shown in Appendix B.4.13 on page 95.

TABLE 39: Mandatory keywords for AIPS SU table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS SU'
NO_IF	I	Number of IFs (n_{IF})
FREQID	I	Frequency setup ID number
VELDEF	A	'RADIO', 'OPTICAL'
VELTYP	A	Velocity coordinate reference

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the source table, this controls the dimension of the flux, frequency offset, and velocity columns.

Frequency setup number. The FREQID column shall contain the frequency setup number of the frequency setup for which the source table applies.

Velocity definition. The VELDEF keyword shall specify whether the velocity definition is radio or optical. The VELTYP keyword shall specify the frame of reference of the velocities. Values like 'GEOVENTR', 'TOPOCENT', 'BARYCENT', 'HELIO', and 'LSR' have been used.

Two new columns, shown below the line, are hereby proposed. AIPS will update all AIPS SU tables to include these columns and will by default fill them with the values in RAEPO and DECEPO. AIPS has kept the pointing position and the phase/delay stopping position separate in single-source and image headers, but, for some reason, has not provided for the columns to allow these parameters to differ during the actual correlation.

TABLE 40: Mandatory and optional columns for the AIPS SU table

Title	Type	Units	Description
ID. NO.	1J		Source number
SOURCE	16A		Source name
QUAL	1J		Source qualifier number
CALCODE	4A		Calibration code
IFLUX	E(n_{IF})	Jy	Stokes I flux
QFLUX	E(n_{IF})	Jy	Stokes Q flux
UFLUX	E(n_{IF})	Jy	Stokes U flux
VFLUX	E(n_{IF})	Jy	Stokes V flux
FREQOFF	D(n_{IF})	Hz	Frequency offset
BANDWIDTH	1D	Hz	Spectral channel separation
RAEPO	1D	degrees	Right ascension of equinox
DECEPO	1D	degrees	Declination of equinox
EPOCH	1D	years	Equinox
RAAPP	1D	degrees	Right ascension of date
DECAPP	1D	degrees	Declination of date
LSRVEL	D(n_{IF})	m/sec	Velocity
RESTFREQ	D(n_{IF})	Hz	Rest frequency
PMRA	1D	deg/day	Proper motion in right ascension
PMDEC	1D	deg/day	Proper motion in declination
RAOBS	1D	degrees	Pointing right ascension of equinox
DECOBS	1D	degrees	Pointing declination of equinox

Source identification. The source shall be uniquely identified by its number specified in the ID. NO. column and also uniquely identified by its the combination of its name, numeric qualifier, and calibration code given in the SOURCE, QUAL, and CALCODE columns, respectively.

Source flux density. The source flux density, if known, shall be specified for each IF in each polarization using the IFLUX, QFLUX, UFLUX, and VFLUX columns.

Frequencies. A source- and IF-dependent frequency offset may be specified in the FREQOFF column and the spectral channel separation is given in the BANDWIDTH column. The former is important in computing the actual observed frequency of each datum, the latter is rarely referenced.

Coordinates. The source right ascension and declination at the chosen equinox are given by the contents of the RAEPO and DECEPO columns while the chosen equinox is given by the EPOCH column. The values of the projected baselines are rotated so that the images are in coordinates of equinox. For reference, the right ascension and declination of date (at the start of the observations) is also given in the RAAPP and DECAPP columns, respectively. AIPS' FITS readers do not assume that these columns are correct and recompute them. They may be used for position shifts if the user wishes to shift in coordinates of date. The right ascension and declination at the chosen equinox at which the antennas pointed are given by the contents of the RAOBS and DECOBS columns. These columns are new and the values from the RAEPO and DECEPO columns will be used if they are absent.

Velocities. The velocity of the Local Standard of Rest with respect to the observer in the direction of the source is given by the values in the LSRVEL column. The rest frequency of the spectral line to be used in velocity computations is specified in the RESTFREQ column. These are both allowed to be functions of IF.

Proper motion. The proper motions of the source in right ascension and declination of equinox, if known, are given in the PMRA and PMDEC columns, respectively.

4.18 AIPS SY SysPower table

The EVLA correlator delivers power spectra without normalization by the autocorrelation self powers. The AIPS SY table contains measurements of the power with the switched noise tubes on and with them off. Using these, plus the noise tube temperatures contained in the AIPS CD table (Section 4.2 on page 15), one may determine a gain by which to correct the un-normalized cross power spectra to produce spectra nearly calibrated in Jy. The AIPS SY table also provides estimates of the system temperature, allowing computation of data weights which are also close to being calibrated in Jy^{-2}. The principles underlying this table have been described by Perley 2010 [22]. An example AIPS SY table header is shown in Appendix B.4.14 on page 97.

TABLE 41: Mandatory keywords for AIPS SY table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS SY'
NO_IF	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
NO_ANT	I	Maximum antenna number

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the SysPower table, this controls the dimensions of the power and gain columns.

Number of polarizations. The AIPS SY table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the NO_POL keyword. In the SysPower table, the value of this keyword controls whether there is one set of data columns or two.

Number of antennas. The value of the NO_ANT keyword shall specify the maximum antenna number to occur in the AIPS SY table.

TABLE 42: Mandatory and optional columns for the AIPS SY table

Title	Type	Units	Description
TIME	1D	days	Central time of interval
TIME INTERVAL	1E	days	Length of interval
SOURCE ID	1J		Source number
ANTENNA NO.	1J		Antenna number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
POWER DIF1	E(n_{IF})	counts	Scaled power on minus power off, polarization 1
POWER SUM1	E(n_{IF})	counts	Scaled power on plus power off, polarization 1
POST GAIN1	E(n_{IF})		Gain after power detection, polarization 1
POWER DIF2	E(n_{IF})	counts	Scaled power on minus power off, polarization 2
POWER SUM2	E(n_{IF})	counts	Scaled power on plus power off, polarization 2
POST GAIN2	E(n_{IF})		Gain after power detection, polarization 2
CAL TYPE	1J		Noise tube type

Time covered by the row. The TIME column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array. The TIME INTERVAL column shall contain the number of days in the time interval represented by the row.

Source identification number. The SOURCE ID column shall contain the source identification number of the source for which the data are valid.

Noise tube type. The CAL TYPE column, if present, specifies whether the normal TCALn noise tubes values from the AIPS CD table are to be used or whether the SOLCALn values are to be used. A value of zero states that normal values are to be used and one that Solar noise tube values are to be used. The column is optional with default value zero.

Antenna and subarray numbers. The ANTENNA NO. column shall contain the antenna identification number of the antenna to which the data apply. The SUBARRAY column shall contain the number of the subarray to which the antenna belongs.

Frequency setup number. The FREQ ID column shall contain the frequency setup number of the frequency setup for which the data apply.

Measured system power. The POWER DIF1 and POWER SUM1 columns contain the difference and sum, respectively, of the system power when the switched noise tube is on and when it is off. They have been scaled by the gains that follow the detection of the system powers. If the value of the NO_POL keyword is two, the POWER DIF2 AND POWER SUM2 columns will contain similar values for the second polarization. The visibilities are scaled as

$$P_{int} = P_{corr} \sqrt{\frac{N_{cal1} N_{cal2}}{P_{dif1} P_{dif2}}} \tag{3}$$

where N_{cali} is the noise tube value for antenna i in the baseline pair and P_{difi} is the content of the AIPS SY table POWER DIFj column. The system temperature of an antenna is given by

$$T_{sys} = N_{cal} P_{sum} / P_{dif}. \tag{4}$$

Post detection gain. The POST GAIN1 column contains the values of the gains which follow the measurement of the total power with the noise tubes on and off. If the value of the NO_POL keyword is two, the POST GAIN2 column contains similar values for the second polarization. The gains have been applied to the values in the POWER SUMi and POWER DIFi columns and so are used only when displaying the actual on and off powers.

4.19 AIPS TY system temperature table

The detected visibilities from most correlators need to be scaled by the measured system temperatures to obtain visibilities in something like Jy. The old VLA correlator delivered visibilities already normalized by the self power. The AIPS TY table contains measurements of the system and antenna temperatures as functions of time, antenna, and IF. In the VLA case, the table was used to convey the applied gains in the TSYS columns and the front-end or back-end system temperature in the TANT columns. AIPS task FILLM was able to remove or apply the gains to produce the desired form of the visibilities for the VLA and to estimate data weights. AIPS task ANCAL is used, primarily for VLBI, to convert measured system temperatures to antenna gains in the AIPS CL table. An example AIPS TY table header is shown in Appendix B.4.15 on page 98.

TABLE 43: Mandatory keywords for AIPS TY table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS TY'
NO_IF	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
REVISION	I	Table format version number (10)

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the SysPower table, this controls the dimensions of the power and gain columns.

Number of polarizations. The AIPS TY table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the NO_POL keyword. In the SysPower table, the value of this keyword controls whether there is one set of data columns or two.

TABLE 44: Mandatory columns for the AIPS TY table

Title	Type	Units	Description
TIME	1D	days	Central time of interval
TIME INTERVAL	1E	days	Length of interval
SOURCE ID	1J		Source number
ANTENNA NO.	1J		Antenna number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
TSYS 1	E(n_{IF})	Kelvin	System temperature, polarization 1
TANT 1	E(n_{IF})	Kelvin	Antenna temperature, polarization 1
TSYS 2	E(n_{IF})	Kelvin	System temperature, polarization 2
TANT 2	E(n_{IF})	Kelvin	Antenna temperature, polarization 2

Time covered by the row. The TIME column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array. The TIME INTERVAL column shall contain the number of days in the time interval represented by the row.

Source identification number. The SOURCE ID column shall contain the source identification number of the source for which the data apply.

Antenna and subarray numbers. The ANTENNA NO. column shall contain the antenna identification number of the antenna to which the data apply. The SUBARRAY column shall contain the number of the subarray to which the antenna belongs.

Frequency setup number. The FREQ ID column shall contain the frequency setup number of the frequency setup for which the data apply.

Measured system temperatures. The TSYS 1 and TANT 1 columns contain the system and antenna temperatures, respectively, for polarization1 as a function of IF. If the value of the NO_POL keyword is two, the TSYS 2 and TANT 2 columns contain similar values for the second polarization. For the old VLA, including the transition time after the ModComps but before Widar, the TSYS *i* columns contained the gain scaling factor for visibilities. Normally, this will have been applied to the data either by the on-line software or by the AIPS FILLM translation task.

4.20 AIPS WX weather table

The AIPS WX table is used to store the measured weather parameters at the antennas during the observing run. These are primarily used for documentation, but may be used to flag data in AIPS task WETHR which also plots the weather table. An example AIPS WX table header is shown in Appendix B.4.16 on page 98.

TABLE 45: Mandatory keywords for AIPS WX table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS WX'
RDATE	D	Reference date
TABREV	I	Table format version number (3)

Reference date. The value of the RDATE parameter will be the date on which the observations begin and from which time is measured in this data set.

TABLE 46: Mandatory and optional columns for the AIPS WX table

Title	Type	Units	Description
TIME	1D	days	Central time of interval
TIME_INTERVAL	1E	days	Length of interval
ANTENNA NO.	1J		Antenna number
SUBARRAY	1J		Subarray number
TEMPERATURE	1E	Centigrade	Surface air temperature
PRESSURE	1E	millibar	Atmospheric surface pressure
DEWPOINT	1E	Centigrade	Dew point temperature
WIND_VELOCITY	1E	m/sec	Steady wind velocity
WIND_DIRECTION	1E	degrees	Wind direction
WIND_GUST	1E	m/sec	Wind gust speed
PRECIPITATON	1E	cm	Precipitation since midnight local time
WVR_H2O	1E	m^{-2}	Water column
IONOS_ELECTRON	1E	m^{-2}	Ion column

AIPS recognizes several other spellings of these column names but they will not be shown here.

Time covered by the row. The TIME column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array. The TIME INTERVAL column shall contain the number of days in the time interval represented by the row.

Antenna and subarray numbers. The ANTENNA NO. column shall contain the antenna identification number of the antenna to which the data apply. A value of 0 means all antennas in the array, which is appropriate for the VLA but not VLBI. The SUBARRAY column shall contain the number of the subarray to which the antenna belongs.

Surface temperature and pressure. The surface temperature and dew point shall be given in the TEMPERATURE and DEWPOINT columns, respectively, in degrees Centigrade. The surface pressure in millibars will be given in the PRESSURE column.

Wind. The wind speed and direction shall be given in the WIND_VELOCITY and WIND_DIRECTION columns, respectively. Units are meters per second and degrees measured East from North. The WIND_GUST column may be used to indicate peak wind gusts as a function of time in meters per second.

Precipitation. The amount of precipitable water accumulated since midnight local time may be recorded in the PRECIPITATION column in cm.

Atmospheric column densities. The column densities of water and electrons above the telescope may be given, if known, in the WVR_H2O and IONOS_ELECTRON columns, respectively.

5 Tables generated by *AIPS* for calibration

5.1 AIPS BD baseline bandpass table

The AIPS BD file carries spectral-channel dependent, baseline based corrections. These corrections will have already been applied to a data set by the task which determined them (BLCHN), but are retained for examination and plotting. This task and form of correction should be viewed as experimental and has seldom been used. Therefore, no example is given in the appendices.

TABLE 47: Mandatory keywords for AIPS BD table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS BD'
NO_IF	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
NO_CHAN	I	Number of spectral channels (n_{chan})
NO_ANT	I	Maximum antenna number

Number of spectral windows and channels. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. The value of the NO_CHAN keyword shall specify the number of spectral channels in the data set. In the baseline bandpass table, the product of these two keyword values controls the dimensions of the real and imaginary columns.

Number of polarizations. The AIPS BD table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the NO_POL keyword. In the baseline bandpass table, the value of this keyword controls whether there is one set of real and imaginary columns or two.

Number of antennas. The value of the NO_ANT keyword shall specify the maximum antenna number to occur in the AIPS BD table.

Normally, the task will determine a single solution for each baseline averaged over all times in the data set. Each row of the table then contains the corrections for a single specified baseline.

TABLE 48: Mandatory columns for the AIPS BD table

Title	Type	Units	Description
TIME	1E	days	Central time of interval
SOURCE ID	1J		Source identification number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
ANTENNA1	1J		First antenna number of baseline
ANTENNA2	1J		Second antenna number of baseline
REAL 1	E(n_{chan},n_{IF})		Real part, polarization 1
IMAG 1	E(n_{chan},n_{IF})		Imaginary part, polarization 1
REAL 2	E(n_{chan},n_{IF})		Real part, polarization 2
IMAG 2	E(n_{chan},n_{IF})		Imaginary part, polarization 2

Time. The TIME column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array.

Source identification number. The SOURCE ID column shall contain the source identification number of the source for which the data apply.

Subarray and frequency setup numbers. The SUBARRAY column shall contain the number of the subarray to which the baseline belongs. The FREQ ID column shall contain the frequency setup number of the frequency setup for which the data apply.

Baseline. The baseline is between two antennas whose numbers are given in the ANTENNA1 and ANTENNA2 columns. Normally, in *AIPS* a baseline is considered to be between the lower numbered and the higher numbered antenna, so the value in the ANTENNA1 column is in general less than that in the ANTENNA2 column.

Correction factors. The real part of the correction for polarization 1 is given in the REAL 1 column while the imaginary part is given in the IMAG 1 column. If the value of the NO_POL keyword is two, the REAL 2 and IMAG 2 columns contain similar values for the second polarization. The complex correction in these columns is divided into the data to correct for closure error.

5.2 AIPS BL baseline correction table

The AIPS BL table is used to correct spectral-channel independent, IF-dependent gains that are a function of baseline and not antenna. In general, these have been found to be independent of time over moderate intervals, *e.g.*, a day's observing. The causes of this closure failure include band-shape mismatch within the bandwidth correlated, a problem particularly present in the VLA with its 50 MHz continuum channel. The correction is determined by dividing the data, calibrated as well as possible using closure gains, by as good a model for the source as may be obtained. If there is not failure of closure, the divided result should have value (1,0) at least when averaged over a significant length of time. The *AIPS* task BLCAL applies the calibration to the data, divides by the model, and then usually average the result over all times. The averaged result is written to the AIPS BL table, which may then be applied by all tasks that apply calibration to data. An example AIPS BL table header is shown in Appendix B.5.1 on page 99.

TABLE 49: Mandatory keywords for AIPS BL table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS BL'
NO_IF	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
NO_ANT	I	Maximum antenna number

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the baseline table, this value controls the dimensions of the real and imaginary columns.

Number of polarizations. The AIPS BL table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the NO_POL keyword. In the baseline table, the value of this keyword controls whether there is one set of real and imaginary columns or two.

Number of antennas. The value of the NO_ANT keyword shall specify the maximum antenna number to occur in the AIPS BL table.

Normally, the task will determine a single solution for each baseline averaged over all times in the data set. Each row of the table then contains the corrections for a single specified baseline. The table was designed to also allow for an additive correction. In *AIPS*, to date, this is always zero. The required columns for the AIPS BL table are shown in Table 50 on the next page.

Time. The TIME column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array.

Source identification number. The `SOURCE ID` column shall contain the source identification number of the source for which the data apply.

Subarray and frequency setup numbers. The `SUBARRAY` column shall contain the number of the subarray to which the baseline belongs. The `FREQ ID` column shall contain the frequency setup number of the frequency setup for which the data apply.

TABLE 50: Mandatory columns for the `AIPS BL` table

Title	Type	Units	Description
TIME	1E	days	Central time of interval
SOURCE ID	1J		Source identification number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
ANTENNA1	1J		First antenna number of baseline
ANTENNA2	1J		Second antenna number of baseline
REAL M1	E(n_{IF})		Real part multiplicative, polarization 1
IMAG M1	E(n_{IF})		Imaginary part multiplicative, polarization 1
REAL A1	E(n_{IF})		Real part additive, polarization 1
IMAG A1	E(n_{IF})		Imaginary part additive, polarization 1
REAL M2	E(n_{IF})		Real part multiplicative, polarization 2
IMAG M2	E(n_{IF})		Imaginary part multiplicative, polarization 2
REAL A2	E(n_{IF})		Real part additive, polarization 2
IMAG A2	E(n_{IF})		Imaginary part additive, polarization 2

Baseline. The baseline is between two antennas whose numbers are given in the `ANTENNA1` and `ANTENNA2` columns. Normally, in *AIPS* a baseline is considered to be between the lower numbered and the higher numbered antenna, so the value in the `ANTENNA1` column is in general less than that in the `ANTENNA2` column.

Multiplicative Correction. The real part of the multiplicative correction for polarization 1 is given in the `REAL M1` column while the imaginary part is given in the `IMAG M1` column. If the value of the `NO_POL` keyword is two, the `REAL M2` and `IMAG M2` columns contain similar values for the second polarization. The complex correction in these columns is divided into the data to correct for closure error.

Additive correction. The real part of the additive correction for polarization 1 is given in the `REAL A1` column while the imaginary part is given in the `IMAG A` column. If the value of the `NO_POL` keyword is two, the `REAL A2` and `IMAG A2` columns contain similar values for the second polarization. The complex correction in these columns is subtracted from the data to correct for closure error. *AIPS* calibration application routines assume that these terms are zero.

5.3 `AIPS BP` bandpass table

The `AIPS BP` table is used to correct for spectral-channel dependent gains which are antenna-dependent, but baseline-independent, *i.e.*, satisfying the standard closure relationships. Traditionally, the `AIPS BP` tables are normalized in some IF-dependent fashion and the assumed to vary slowly, if at all, with time. The more rapid gain variations are then assumed to be only spectral-window dependent and are left to the `AIPS CL` table. The `AIPS BP` table does not require this tradition, although considerations of signal-to-noise ratios might. An example `AIPS BP` table header is shown in Appendix B.5.2 on page 100.

TABLE 51: Mandatory and useless keywords for AIPS BP table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS BP'
NO_IF	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
NO_CHAN	I	Number of values in BP (n_{BPchn})
NO_ANT	I	Maximum antenna number
START_CHAN	I	Start channel number of BP values
NO_SHFTS	I	Code for source of BP values
BP_TYPE	A	' ', 'CHEBSHEV' recognized
LOW_SHFT	E	Most negative shift
SHFT_INC	E	Shift increment

Number of spectral windows and bandpass values. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. The value of the NO_CHAN keyword shall specify the number of values in the BP table. In the bandpass table, the product of these two keyword values controls the dimensions of the real and imaginary columns. The number of spectral windows controls the dimensions of the channel shift and weight columns as well. The value of the NO_CHAN keyword may be (usually is) the number of spectral channels in the data set per spectral window. It may be smaller and it may represent the number of terms on a Chebyshev polynomial expansion of the "normal" bandpass function.

Number of polarizations. The AIPS BD table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the NO_POL keyword. In the baseline bandpass table, the value of this keyword controls whether there is one set of reference antennas, weights, real, and imaginary columns or two.

Number of antennas. The value of the NO_ANT keyword shall specify the maximum antenna number to occur in the AIPS BP table.

Starting channel. The table format allows the storage of a partial bandpass function beginning at the channel number given by the value of the START_CHAN keyword and continuing for a total of the value of the NO_CHAN keyword channels. The assumption was that it would be worthwhile to keep only those channels which are likely to be usable in the bandpass table. Recommended value is 1 with all channels in the data set represented in the BP table. If channels are to be discarded permanently, do that over the full data set with a copy function.

Data used to determine the bandpass. The value of the NO_SHFTS keyword gives the type of data used to compute the bandpass. A value of 1 means only cross-power data, 2 means only total power (autocorrelation) data, and 3 or 0 means both or unspecified. The value of the keyword is set by *AIPS*, but not used.

Bandpass type. The BP_TYPE keyword is used to indicate the type of bandpass function stored in the BP table. Blank is taken be a "normal" bandpass, *i.e.*, the full one complex number to be applied to one visibility. Any thing else, and 'CHEBSHEV' is to be preferred, implies that the values in the table are the parameters of a Chebyshev expansion of the normal bandpass function. *AIPS* will still handle that, but has been changed internally, to do the expansion in the task that does that fitting (CPASS) so that normal plotting and averaging functions may be done on it.

Shifts. There was once a concept to store and use various shifted bandpass functions apparently. No real documentation or use of the LOW_SHFT and SHFT_INC keywords can be found. They are therefore optional.

Each row of the table contains a bandpass solution for one antenna over a specified time interval. Other solutions for that antenna, at other times, may also appear in the table. Table 52 on the next page shows the required columns of the AIPS BP table.

Time. The TIME column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array. The INTERVAL shall include the length of time over which data were averaged to produce the solution given in the row.

Source identification number. The SOURCE ID column shall contain the source identification number of the source for which the data apply.

Subarray and frequency setup numbers. The SUBARRAY column shall contain the number of the subarray to which the baseline belongs. The FREQ ID column shall contain the frequency setup number of the frequency setup for which the data apply.

Antenna. The ANTENNA column will contain the identification number of the antenna to which the bandpass correction applies.

Channel separation. The channel separation in Hz is given in the BANDWIDTH column. It is not actually used.

Channel shifts. The CHN_SHIFT column was used to record the shifting of bandpasses required to align the bandpasses for the VBLA. It was since realized that one has to do that another way, during bandpass application, and so this column no longer has value.

TABLE 52: Mandatory columns for the AIPS BP table

Title	Type	Units	Description
TIME	1D	days	Central time of interval
INTERVAL	1E	days	Length of interval
SOURCE ID	1J		Source identification number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
ANTENNA	1J		Antenna number
BANDWIDTH	1E	Hz	Spectral channel separation
CHN_SHIFT	D(n_{IF})		VLBA channel shift
REFANT 1	1J		Reference antenna, polarization 1
WEIGHT 1	E(n_{IF})		Solution weight, polarization 1
REAL 1	E(n_{chan},n_{IF})		Real part, polarization 1
IMAG 1	E(n_{chan},n_{IF})		Imaginary part, polarization 1
REFANT 2	1J		Reference antenna, polarization 2
WEIGHT 2	E(n_{IF})		Solution weight, polarization 2
REAL 2	E(n_{chan},n_{IF})		Real part, polarization 2
IMAG 2	E(n_{chan},n_{IF})		Imaginary part, polarization 2

Reference antenna. The REFANT 1 column shall give the identification number of the antenna used as a phase reference for this bandpass solution in polarization 1 for all IFs. If the value of the NO_POL keyword is two, the REFANT 2 column contains a similar value for the second polarization.

Weights. The WEIGHT 1 column shall give the relative weight of this bandpass solution in polarization 1 for each IF. If the value of the NO_POL keyword is two, the WEIGHT 2 column contains similar values for the second polarization.

Correction factors. The real part of the correction for polarization 1 is given in the REAL 1 column while the imaginary part is given in the IMAG 1 column. If the value of the NO_POL keyword is two, the REAL 2 and IMAG 2 columns contain similar values for the second polarization. For each baseline, the product of the complex correction of the first antenna and the complex conjugate of the correction for the second antenna is divided into the data to correct for the shape of the bandpass and potentially for overall changes in amplitude and phase.

5.4 AIPS BS baseline fringe solution table

The AIPS BS table is used to store solutions for residual group delay and phase rate and acceleration performed on a baseline basis. The values in the table may be converted to antenna-based values and applied to an AIPS CL table. The tasks in AIPS which perform these two functions are BLING and BLAPP.

Mk4 VLBI data files may also contain information which is converted by `MK4IN` into an `AIPS BS` table. Because this table does not appear to be widely used, no example of its header is given.

TABLE 53: Mandatory keywords for `AIPS BS` table headers

Keyword	Value type	Value
`EXTNAME`	A	`'AIPS BS'`
`NO_IF`	I	Number of IFs (n_{IF})
`MODE`	A	Solution mode used to derive table
`VERSION`	I	Table revision number (4)

Number of spectral windows. The value of the `NO_IF` keyword shall specify the number of spectral windows (IFs) in the data set. In the baseline table, this value controls the dimensions of the real and imaginary columns.

Solution mode. The value of the `MODE` keyword shall be a string identifying the parameters for which solutions were obtained. Recognized values include (1) `'INDE'` for independent delays in each IF, (2) `'VLBA'` for one delay for all IFs, (3) `'MK3'` for multi-band and single-band delays, and (4) `'RATE'` for rates only.

TABLE 54: Mandatory columns for the `AIPS BS` table

Title	Type	Units	Description
`TIME`	`1D`	days	Central time of interval
`TIME INTERVAL`	`1E`	days	Length of interval
`BASELINE`	`2J`		Identification numbers of antennas in baseline
`SUBARRAY`	`1J`		Subarray number
`STOKES`	`1J`		Stokes value of data used
`SOURCE`	`1J`		Source identification number
`VECTOR AMPLITUDE`	E(n_{IF})	Jy	Peak amplitude fit
`SCALAR AMPLITUDE`	E(n_{IF})	Jy	Scalar average amplitude
`RESIDUAL MB DELAY`	`1E`	seconds	Multi-band delay
`MB DELAY ERROR`	`1E`	seconds	Multi-band delay error
`MB DELAY AMBIGUITY`	`1E`	seconds	Multi-band delay ambiguity
`RESIDUAL SB DELAY`	E(n_{IF})	seconds	Single-band delay
`SB DELAY ERROR`	E(n_{IF})	seconds	Single band delay error
`SB DELAY AMBIGUITY`	`E`	seconds	Single-band delay ambiguity
`RESIDUAL FRINGE RATE`	E(n_{IF})	Hz	Single-band fringe rate
`FRINGE RATE ERROR`	E(n_{IF})	Hz	Single-band fringe rate error
`FRINGE RATE AMBIGUITY`	`E`	Hz	Single-band fringe rate ambiguity
`RESIDUAL ACCELERATION`	E(n_{IF})	Hz/sec	Rate of change of single-band fringe rate
`ACCELERATION ERROR`	E(n_{IF})	Hz/sec	Uncertainty in rate of change of fringe rate
`RESIDUAL PHASE`	E(n_{IF})	degrees	Fit phase
`PHASE ERROR`	E(n_{IF})	degrees	Uncertainty in fit phase

Time. The `TIME` column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array. The `TIME INTERVAL` shall include the length of time over which data were averaged to produce the solution given in the row.

Baseline. The `BASELINE` column shall contain 2 positive integers per row which are the identification numbers of the two antennas in the baseline to which the solution data apply.

Subarray number. The `SUBARRAY` column shall contain the number of the subarray to which the baseline belongs.

Stokes. The `STOKES` column shall contain the conventional value for the Stokes parameter of the data used to obtain the solution. Polarization values 1 through 4 are used for I, Q, U, and V, respectively, whilst values -1 through -4 are used for polarization products RR, LL, RL, and LR, respectively.

Source identification number. The SOURCE column shall contain the source identification number of the source for which the data apply.

Amplitudes. The simple average of the fringe amplitudes in the data is given in the SCALAR AMPLITUDE while the peak amplitude found in the fringe fit is given in the VECTOR AMPLITUDE column. The ratio of the vector to scalar averages is an indication of the signal-to-noise in the solution.

Multi-band delay solutions. The multi-band residual delay solution, uncertainty, and possible ambiguity are given in the RESIDUAL MB DELAY, MB DELAY ERROR, and MB DELAY AMBIGUITY columns, respectively. The ambiguity is one over the greatest common denominator of the list of IF frequency differences. If multi-band delays have not been found, these columns will be filled with IEEE NaNs (not-a-number).

Single-band delay solutions. The single-band residual delay solution, uncertainty, and possible ambiguity are given in the RESIDUAL SB DELAY, SB DELAY ERROR, and SB DELAY AMBIGUITY columns, respectively. There is a value for each spectral window. The ambiguity is one over the channel separation. If single-band delays have not been found, these columns will be filled with IEEE NaNs (not-a-number).

Fringe rate. The single-band residual fringe rate solution, uncertainty, and possible ambiguity are given in the RESIDUAL FRINGE RATE, FRINGE RATE ERROR, and FRINGE RATE AMBIGUITY columns, respectively. There is a value for each spectral window. The ambiguity is one over the time separating data samples. If fringe rates have not been found, these columns will be filled with IEEE NaNs (not-a-number).

Residual acceleration. The rate of change of fringe rate and its uncertainty are given in the RESIDUAL ACCELERATION and ACCELERATION ERROR columns, respectively. If fringe-rate accelerations have not been found, these columns will be filled with IEEE NaNs (not-a-number).

Residual phase. The residual phase and its uncertainty are given in the RESIDUAL PHASE and PHASE ERROR columns, respectively.

5.5 AIPS CL calibration table

The AIPS CL table contains the complex gains to be applied to the data in a spectroscopic channel-independent, IF-dependent basis as a function of the time of observation. AIPS CL tables are found only in multi-source *uv* data sets. They contain information about corrections made during the correlation process and information developed during the data processing inside *AIPS*. The tables are initially constructed to have a time sample at the beginning and end of every "scan" as well as at regular intervals within each scan. The definition of scan is left up to the observer who scheduled the telescope and the interval within the scan is selected by the user of the program that creates the CL table. All possible sources of complex gain are eventually entered into this table. Opacity and antenna gain elevation dependencies may be entered during the initial table creation or by later tasks. Corrections for total power may be entered from the AIPS SY or AIPS TY tables. Calibration tasks generally write their results in a solution (AIPS SN) table which is then interpolated onto the AIPS CL table. CLCOR modifies the AIPS CL table directly. An example AIPS CL table header is shown in Appendix B.5.3 on page 102.

TABLE 55: Mandatory keywords for AIPS CL table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS CL'
NO_IF	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
NO_ANT	I	Maximum antenna number
NO_TERM	I	Number terms in geometric delay polynomial
MGMOD	E	Overall scale factor to apply
REVISION	I	Table format revision number (10)

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the baseline table, this value controls the dimensions of the real and imaginary columns.

Number of polarizations. The AIPS CL table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the NO_POL keyword. In the calibration table, the value of this keyword controls whether there is one set of calibration columns or two.

Number of antennas. The value of the NO_ANT keyword shall specify the maximum antenna number to occur in the AIPS CL table.

Model polynomial. The value of the NO_TERM keyword shall specify the number of terms in the model polynomial for geometric delay.

Overall gain factor. The value of the MGMOD keyword shall specify a factor which must be applied to all visibility amplitudes in addition to the complex gains specified in the table rows. This is a quick way to adjust the overall scale, but is independent of spectral window.

TABLE 56: Mandatory columns for the AIPS CL table

Title	Type	Units	Description
TIME	1D	days	Central time of interval
TIME INTERVAL	1E	days	Central time of interval
SOURCE ID	1J		Source identification number
ANTENNA	1J		Antenna number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
I.FAR.ROT	1E	radians m^{-2}	Ionospheric Faraday rotation
GEODELAY	D(n_{poly})	seconds	Geometric delay polynomial
DOPOFF	E(n_{IF})	Hz	Correction to observing frequency
ATMOS	1E	seconds	Atmospheric delay
DATMOS	1E	sec/sec	Rate of change of atmospheric delay
MBDELAY1	1E	seconds	Multi-band delay, polarization 1
CLOCK 1	1E	seconds	Clock error, polarization 1
DCLOCK 1	1E	sec/sec	rate of change of clock error, polarization 1
DISP 1	1E	sec m^{-2}	Dispersive delay, polarization 1
DDISP 1	1E	s/s m^{-2}	Rate of change of dispersive delay, polarization 1
REAL1	E(n_{IF})		Real part of complex gain correction, polarization 1
IMAG1	E(n_{IF})		Imaginary part of complex gain correction, polarization 1
RATE 1	E(n_{IF})	sec/sec	Single-band rate correction, polarization 1
DELAY 1	E(n_{IF})	seconds	Single-band delay correction, polarization 1
WEIGHT 1	E(n_{IF})		Solution weight, polarization 1
REFANT 1	J(n_{IF})		Reference antenna, polarization 1
MBDELAY2	1E	seconds	Multi-band delay, polarization 2
CLOCK 2	1E	seconds	Clock error, polarization 2
DCLOCK 2	1E	sec/sec	rate of change of clock error, polarization 2
DISP 2	1E	sec m^{-2}	Dispersive delay, polarization 2
DDISP 2	1E	s/s m^{-2}	Rate of change of dispersive delay, polarization 2
REAL2	E(n_{IF})		Real part of complex gain correction, polarization 2
IMAG2	E(n_{IF})		Imaginary part of complex gain correction, polarization 2
RATE 2	E(n_{IF})	sec/sec	Single-band rate correction, polarization 2
DELAY 2	E(n_{IF})	seconds	Single-band delay correction, polarization 2
WEIGHT 2	E(n_{IF})		Solution weight, polarization 2
REFANT 2	J(n_{IF})		Reference antenna, polarization 2

When the AIPS CL table is applied to uncalibrated visibilities, the values in the REALi, IMAGi, DELAY i, and RATE i columns are used. The ionospheric Faraday rotation correction is applied to the phases of the cross-hand data scaled by the wavelength squared and the dispersion is applied to all phases as the wavelength times the difference in dispersions of the two antennas. Other columns containing corrections (atmosphere, clock, multi-band delay) are ignored. It is assumed that any complex gain, delay, and rate corrections due to these factors have either already been applied to the data or have been applied to the gain, delay, and rate values in the table.

Time. The `TIME` column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array. The `TIME INTERVAL` column shall contain the length of time to which this row applies. Calibrations are normally interpolated between values of the `TIME` column rather than being taken as step functions over the interval.

Source identification number. The `SOURCE ID` column shall contain the source identification number of the source for which the data apply.

Antenna. The identification number of the antenna for which the row applies is given by the value in the `ANTENNA` column. All calibrations in the `AIPS CL` table are antenna based.

Subarray and frequency setup numbers. The `SUBARRAY` column shall contain the number of the subarray to which the baseline belongs. The `FREQ ID` column shall contain the frequency setup number of the frequency setup for which the data apply.

Ionospheric Faraday rotation. Corrections to be made to the visibilities for ionospheric Faraday rotation are given in the `I.FAR.ROT` column. They are applied to the phases of the cross-hand polarizations. Corrections applied to the visibilities by the correlator are ignored in creating `AIPS CL` table 1.

Geometric delay. The geometric delay model used by the correlator is given as n_{poly} (the value of the `NO_TERM` keyword) values of a polynomial expansion in time (from the value given in the `TIME` column) in the `GEODELY` column.

Doppler offset. The `DOPOFF` column conveys n_{IF} values giving the actual observed frequency minus the time-independent frequency for each IF. The time-independent frequencies are described for each IF and frequency setup number in the `AIPS FQ` table (Section 4.7 on page 23) plus the frequency offsets described for each source in the `AIPS SU` (Section 4.17 on page 35) table plus the reference frequency given in the *uv*-data header.

Atmospheric delay. Any atmospheric delay correction entered by the correlator into the initial visibilities plus any entered into the complex gain, delay, and rate of the `CL` table by *AIPS* is given in the `ATMOS` column. The rate of change of this correction is given in the `DATMOS` column.

Multi-band delays. The multi-band delay is a residual delay common to all IFs. Any such delay applied to the visibilities or the single-band delays of polarization 1 shall be recorded in the `MBDELAY1` column. If the value of the `NO_POL` keyword is two, the `MBDELAY2` column shall contain similar values for the second polarization. Note that there is assumption that the multi-band delay shall have already been applied to the data or shall have been added to the single-band delays of every IF.

Clock errors. Any clock error correction entered by the correlator into the initial visibilities of polarization 1 plus any entered into the complex gain, delay, and rate of the `CL` table by *AIPS* is given in the `CLOCK 1` column. The rate of change of this correction is given in the `DCLOCK 1` column. If the value of the `NO_POL` keyword is two, the `CLOCK 2` and `DCLOCK 2` columns contain similar values for the second polarization.

Dispersive delays. Any corrections to polarization 1 for dispersive delays to be made to the data are entered into the `DISP 1` column. The rate of change of this correction is given in the `DDISP 1` column. If the value of the `NO_POL` keyword is two, the `DISP 2` and `DDISP 2` columns contain similar values for the second polarization. *AIPS* corrects the data for the values in these columns, so any values applied by the correlator are discarded in creating `AIPS CL` table 1.

Complex gains. The complex gain correction of polarization 1 is given in the `REAL1` and `IMAG1` columns for the real and imaginary parts, respectively. If the value of the `NO_POL` keyword is two, the `REAL2` and `IMAG2` columns contain similar values for the second polarization. The product of the complex gain of antenna i with the conjugate of the complex gain of antenna j is divided into the uncalibrated visibility from baseline ij to produce the calibrated visibility. Failed solutions will appear as IEEE NaNs (not-a-number) in the FITS file.

Delays. The data of polarization 1 are also to be corrected for residual delay error given in the `DELAY 1` column and rate of change of the residual delay error given in the `RATE 1` column. If the value of the `NO_POL` keyword is two, the `DELAY 2` and `RATE 2` columns contain similar values for the second polarization. Failed solutions will appear as IEEE NaNs (not-a-number) in the FITS file.

Solution weights. The relative reliability of the solutions for complex gain of polarization 1 are indicated by the values contained in the `WEGHT 1` column, with values ≤ 0 indicating failure. If the value of the `NO_POL` keyword is two, the `WEIGHT 2` column contains similar values for the second polarization.

Reference antenna(s). The index number of the phase reference antenna(s) used for the complex gains of polarization 1 are indicated by the values contained in the `REFANT 1` column. If the value of the `NO_POL` keyword is two, the `REFANT 2` column contains similar values for the second polarization.

5.6 `AIPS CP` source polarization spectrum table

The `AIPS CP` table is used to save the polarization spectrum fit to polarization calibration sources by `PCAL`. It may be displayed by `POSSM`. An example `AIPS CP` table header is shown in Appendix B.5.4 on page 104.

TABLE 57: Mandatory keywords for `AIPS CP` table headers

Keyword	Value type	Value
`EXTNAME`	A	`'AIPS CP'`
`NO_IF`	I	Number of IFs (n_{IF})
`NO_CHAN`	I	Number of spectral channels (n_{chan})
`FREQID`	I	Frequency setup number

Number of spectral windows and spectral channels. The value of the `NO_IF` keyword shall specify the number of spectral windows (IFs) in the data set. The value of the `NO_CHAN` keyword shall specify the number of spectral channels in the data set. In the polarization spectrum table, the product of these two keyword values controls the dimensions of the flux columns.

Frequency setup number. The `FREQID` column shall contain the frequency setup number of the frequency setup for which the source polarization table applies.

TABLE 58: Mandatory columns for the `AIPS CP` table

Title	Type	Units	Description
`SOURCE`	`16A`		Source name
`SOURCE ID`	`1J`		Source identification number
`I`	`E`(n_{chan},n_{IF})	Jy	Source flux, I polarization
`Q`	`E`(n_{chan},n_{IF})	Jy	Source flux, Q polarization
`U`	`E`(n_{chan},n_{IF})	Jy	Source flux, U polarization
`V`	`E`(n_{chan},n_{IF})	Jy	Source flux, V polarization

Source. The source alphabetic name is given in the `SOURCE` column and the identification number used for that source is given in the `SOURCE ID` column.

Stokes polarization fluxes. The spectrum of the source fluxes is given in the `I`, `Q`, `U`, and `V` columns for Stokes values I, Q, U, and V, respectively.

5.7 `AIPS GP` GPS table

The `AIPS GP` table contains the total electron content (TEC) data from a GPS satellite as a function of time. The contents have been corrected for receiver and satellite delay offsets. For antennas situated near the GPS receiver, corrections for the ionospheric Faraday rotation and delay may be made. The *AIPS* tasks are LDGPS to load an ASCII GPS file into an AIPS GP table and APGPS to apply it. APGPS changes the Faraday rotation and dispersive delay parameters in the AIPS CL table. These are then applied whenever the data are calibrated. p

TABLE 59: Mandatory keywords for AIPS GP table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS GP'
RECVR	A	Rceiver name
RLONG	E	Receiver longitude (degrees)
RLAT	E	Receiver latitude (degrees)
RHEIGHT	E	Receiver elevation (meters)
VERSION	I	Table version number (1)

GPS receiver station. The receiver station name, longitude, latitude, and height above MSL are specified in the RECVR, RLONG, RLAT¡ and RHEIGHT keywords, respectively.

TABLE 60: Mandatory columns for the AIPS GP table

Title	Type	Units	Description
TIME	1D	days	Central time of interval
PRN	1J		Satellite ID number
AZIMUTH	1E	degrees	Satellite azimuth
ELEVATION	1E	degrees	Satellite elevation
TEC FROM DELAY	1E	m^{-2}	Total electron content from delay
TEC FROM PHASE	1E	m^{-2}	Total electron content from phase

Time. The TIME column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array.

Satellite. The satellite identification number ("PRN") and its azimuth and elevation from the receiver station are recorded in the PRN, AZIMUTH, and ELEVATION columns, respectively.

Total electron content. The total electron content in electrons per meter^2 deduced from GPS delay and phase measurements are reported in the TEC FROM DELAY and TEC FROM PHASE columns, respectively.

5.8 AIPS PD polarization D-term spectrum table

The AIPS PD table is used the store the polarization "D-terms" as a function of spectral channel and spectral window as found by AIPS task PCAL. The calibration routines apply the AIPS PD table to the data if the user requests polarization calibration. If an AIPS PD table is not present, then spectral-channel independent D-terms are found in the AIPS AN table. An example AIPS PD table header is shown in Appendix B.5.5 on page 104.

TABLE 61: Mandatory keywords for AIPS PD table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS PD'
NO_IF	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
NO_CHAN	I	Number of spectral channels (n_{chan})
NO_ANT	I	Maximum antenna number
POLTYPE	A	Type of solution

Number of spectral windows and channels. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. The value of the NO_CHAN keyword shall specify the number of spectral channels in the data set. In the polarization correction bandpass table, the product of these two keyword values controls the dimensions of the real and imaginary columns.

Number of polarizations. The AIPS PD table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the NO_POL keyword. In the polarization correction bandpass table, the value of this keyword controls whether there is one set of real and imaginary columns or two.

Number of antennas. The value of the NO_ANT keyword shall specify the maximum antenna number to occur in the AIPS PD table.

Polarization parametrization. If the table contains information about the polarization characteristics of the feeds, then the feed parametrization that is used shall be indicated by the value of the POLTYPE keyword, as given in Table 9 page 13.

TABLE 62: Mandatory columns for the AIPS PD table

Title	Type	Units	Description
ANTENNA	1J		Antenna number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
REFANT	1J		Antenna number of reference antenna
P_DIFF	E(n_{chan},n_{IF})		R-L phase difference in 'ORI-ELP'
REAL 1	E(n_{chan},n_{IF})		D-term real part polarization 1
IMAG 1	E(n_{chan},n_{IF})		D-term imaginary part polarization 1
REAL 2	E(n_{chan},n_{IF})		D-term real part polarization 2
IMAG 2	E(n_{chan},n_{IF})		D-term imaginary part polarization 2

Antenna. The identification number of the antenna for which the row applies is given by the value in the ANTENNA column. All calibrations in the AIPS PD table are antenna based.

Subarray and frequency setup numbers. The SUBARRAY column shall contain the number of the subarray to which the baseline belongs. The FREQ ID column shall contain the frequency setup number of the frequency setup for which the data apply.

Reference antenna. One antenna is taken to be unpolarized in the right-hand polarization. That antenna is recorded in the REFANT column.

Phase difference. A right minus left phase difference is fit in the 'ORI=ELP' polarization model. Its values are recorded in the P_DIFF column.

D-term fit. The complex D-term fit by the particular polarization parametrization for polarization 1 is stored in the REAL 1 and IMAG 1 columns for the real and imaginary parts, respectively. If the value of the keyword NO_POL is 2 and it really has to be to solve for polarization, the REAL 2 and IMAG 2 columns contain the D-terms for the second polarization.

5.9 AIPS PP polarization phase difference spectrum table

The AIPS PP table is used the store the correction in phase produced by RLDIF and XYDIF. The values are to be added to the second polarization in the bandpass (BP) table and also the antenna D-term (PD) and source polarization (CP) spectrum tables. This addition may be done by RLDIF and XYDIF, but may be deferred for later or for use on a different data set. Task PPAPP may be used to apply the correction and POSSM may be used to plot the PP spectrum. An example AIPS PP table header is shown in Appendix B.5.6 on page 105.

TABLE 63: Mandatory keywords for AIPS PP table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS PP'
NO_IF	I	Number of IFs (n_{IF})
NO_CHAN	I	Number of spectral channels (n_{chan})
BIF	A	Beginning IF in RLDIF
BCHAN	A	Beginning spectral channel in RLDIF

Number of spectral windows and channels. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the PP data. The value of the NO_CHAN keyword shall specify the number of spectral channels in those data. In the polarization phase correction bandpass table, the product of these two keyword values controls the dimensions of the real and imaginary columns.

Initial spectral channel. The AIPS PP table may not contain data for all spectral windows and spectral channels. Keyword BIF specifies the initial spectra window and keyword BCHAN specifies the initial spectral channel in each spectral window for the data contained in the table.

TABLE 64: Mandatory columns for the AIPS PP table

Title	Type	Units	Description
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
PHASE DIFF	E(n_{chan},n_{IF})		R-L phase difference in degrees

Subarray and frequency setup numbers. The SUBARRAY column shall contain the number of the subarray to which the baseline belongs. The FREQ ID column shall contain the frequency setup number of the frequency setup for which the data apply.

Phase difference. A right minus left phase difference correction is stored in degrees in the PHASE DIFF column.

5.10 AIPS SN solution table

The AIPS SN table is used to store the results of calibration fits to data. The complex gains, delays, and rates are stored at times present in the, possibly time averaged, calibration observations. These irregularly spaced data are then smoothed, edited, and interpolated to the times in the AIPS CL table by *AIPS* task CLCAL. For single-source files, the AIPS SN data are interpolated to the times of the data being calibrated and applied. An example AIPS SN table header is shown in Appendix B.5.7 on page 106.

TABLE 65: Mandatory keywords for AIPS SN table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS SN'
NO_IF	I	Number of IFs (n_{IF})
NO_POL	I	Number of polarizations
NO_ANT	I	Maximum antenna number
MGMOD	E	Overall scale factor to apply
APPLIED	L	Has table been applied to a CL table
REVISION	I	Table format revision number (10)
NO_NODES	I	Number RA and Dec interpolation nodes (≤ 25)
RA_OFFi	E	RA offset interpolation node, $i = 1$,NO_NODES
DEC_OFFi	E	Dec offset interpolation node, $i = 1$,NO_NODES

Number of spectral windows. The value of the NO_IF keyword shall specify the number of spectral windows (IFs) in the data set. In the baseline table, this value controls the dimensions of the real and imaginary columns.

Number of polarizations. The AIPS SN table may contain information for one or two orthogonal polarizations. The number of polarizations shall be given by the NO_POL keyword. In the calibration table, the value of this keyword controls whether there is one set of calibration columns or two.

Number of antennas. The value of the NO_ANT keyword shall specify the maximum antenna number to occur in the AIPS SN table.

Overall gain factor. The value of the MGMOD keyword shall specify a factor which must be applied to all visibility amplitudes in addition to the complex gains specified in the table rows. This is a quick way to adjust the overall scale, but is independent of spectral window.

Table application. If an AIPS SN table has been applied to an AIPS CL the value of the APPLIED keyword shall be true. Otherwise it will be false. The table may be applied more than once, but only with great care, and never applied to a table to which it has already been applied.

Interpolation nodes. The number of interpolation nodes shall be specified by the value of the keyword NO_NODES. If this value is greater than 0, additional keywords shall be present and have values of the right ascension and declination offsets (in degrees) for each node. *AIPS* seems always to use zero nodes, making it hard to determine the actual purpose of this function.

TABLE 66: Mandatory columns for the AIPS SN table

Title	Type	Units	Description
TIME	1D	days	Central time of interval
TIME INTERVAL	1E	days	Central time of interval
SOURCE ID	1J		Source identification number
ANTENNA NO.	1J		Antenna number
SUBARRAY	1J		Subarray number
FREQ ID	1J		Frequency setup ID number
I.FAR.ROT	1E	radians m^{-2}	Ionospheric Faraday rotation
NODE NO.	1J		Interpolation node number
MBDELAY1	1E	seconds	Multi-band delay, polarization 1
DISP 1	1E	sec m^{-2}	Dispersive delay, polarization 1
DDISP 1	1E	s/s m^{-2}	Rate of change of dispersive delay, polarization 1
REAL1	E(n_{IF})		Real part of complex gain correction, polarization 1
IMAG1	E(n_{IF})		Imaginary part of complex gain correction, polarization 1
DELAY 1	E(n_{IF})	seconds	Single-band delay correction, polarization 1
RATE 1	E(n_{IF})	sec/sec	Single-band rate correction, polarization 1
WEIGHT 1	E(n_{IF})		Solution weight, polarization 1
REFANT 1	J(n_{IF})		Reference antenna, polarization 1
MBDELAY2	1E	seconds	Multi-band delay, polarization 2
DISP 2	1E	sec m^{-2}	Dispersive delay, polarization 2
DDISP 2	1E	s/s m^{-2}	Rate of change of dispersive delay, polarization 2
REAL2	E(n_{IF})		Real part of complex gain correction, polarization 2
IMAG2	E(n_{IF})		Imaginary part of complex gain correction, polarization 2
RATE 2	E(n_{IF})	sec/sec	Single-band rate correction, polarization 2
DELAY 2	E(n_{IF})	seconds	Single-band delay correction, polarization 2
WEIGHT 2	E(n_{IF})		Solution weight, polarization 2
REFANT 2	J(n_{IF})		Reference antenna, polarization 2

When the AIPS SN table is applied to uncalibrated visibilities, the values in the REALi, IMAGi, DELAY i, and RATE i columns are used. The ionospheric Faraday rotation correction is applied to the phases of the cross-hand data. The AIPS SN table does not contain dispersion columns. Other columns containing corrections (multi-band delay) are ignored. It is assumed that any complex gain, delay, and rate corrections due to the multi-band delays have either already been applied to the data or have been applied to the gain, delay, and rate values in the table.

Time. The TIME column shall contain the central time covered by the current row as the number of days that have elapsed since 0 hours on the reference date in the time system used for the array. The TIME INTERVAL column shall contain the length of time to which this row applies. Calibrations are normally interpolated between values of the TIME column rather than being taken as step functions over the interval.

Source identification number. The SOURCE ID column shall contain the source identification number of the source for which the data apply.

Antenna. The identification number of the antenna for which the row applies is given by the value in the ANTENNA column. All calibrations in the AIPS CL table are antenna based.

Subarray and frequency setup numbers. The SUBARRAY column shall contain the number of the subarray to which the baseline belongs. The FREQ ID column shall contain the frequency setup number of the frequency setup for which the data apply.

Ionospheric Faraday rotation. Corrections to be made to the visibilities are given in the I.FAR.ROT column. They are applied to the phases of the cross-hand polarizations. CLCAL does not currently update this column in the AIPS CL from the values in the AIPS SN table.

Interpolation node number. If multiple interpolation nodes are used, the value of the node used to obtain the gains in the row shall be contained in the NODE NO. column.

Multi-band delays. The multi-band delay is a residual delay common to all IFs. Any such delay applied to the visibilities or the single-band delays of polarization 1 shall be recorded in the MBDELAY1 column. If the value of the NO_POL keyword is two, the MBDELAY2 column shall contain similar values for the second polarization. Note that there is assumption that the multi-band delay shall have already been applied to the data or shall have been added to the single-band delays of every IF. CLCAL currects the multi-band delays for rates and then adds the AIPS SN table interpolated values to the comparable columns in the AIPS CL table.

Dispersive delays. Any corrections to polarization 1 for dispersive delays to be made to the data are entered into the DISP 1 column. The rate of change of this correction is given in the DDISP 1 column. If the value of the NO_POL keyword is two, the DISP 2 and DDISP 2 columns contain similar values for the second polarization.

Complex gains. The complex gain correction of polarization 1 is given in the REAL1 and IMAG1 columns for the real and imaginary parts, respectively. If the value of the NO_POL keyword is two, the REAL2 and IMAG2 columns contain similar values for the second polarization. The product of the complex gain of antenna i with the conjugate of the complex gain of antenna j is divided into the uncalibrated visibility from baseline ij to produce the calibrated visibility. Failed solutions will appear as IEEE NaNs (not-a-number) in the FITS file.

Delays. The data of polarization 1 are also to be corrected for residual delay error given in the DELAY 1 column and rate of change of the residual delay error given in the RATE 1 column. If the value of the NO_POL keyword is two, the DELAY 2 and RATE 2 columns contain similar values for the second polarization. Failed solutions will appear as IEEE NaNs (not-a-number) in the FITS file.

Solution weights. The relative reliability of the solutions for complex gain of polarization 1 are indicated by the values contained in the WEIGHT 1 column, with values ≤ 0 indicating failure. If the value of the NO_POL keyword is two, the WEIGHT 2 column contains similar values for the second polarization.

Reference antenna(s). The index number of the phase reference antenna(s) used for the complex gains of polarization 1 are indicated by the values contained in the REFANT 1 column. If the value of the NO_POL keyword is two, the REFANT 2 column contains similar values for the second polarization.

6 Tables generated by *AIPS* for imaging

6.1 AIPS CC Clean components table

The AIPS CC table is used to list the components of a source model. The model may consist of point sources, Gaussians, convolved Gaussians, and uniform optically-thin spheres. There is no requirement that all components in an AIPS CC file be of the same type or dimension. However, the gridded modeling routines cannot handle AIPS CC files of mixed type or even of mixed model component dimensions. Source models are produced during imaging with deconvolution (*AIPS* task IMAGR) and by modeling tasks such as JMFIT, SAD, FACES, and others. They are used to compare models with data in self-calibration (*AIPS* task CALIB) and numerous other calibration tasks. *AIPS* will accept and use AIPS CC tables attached to uv data files as well as to images. An example AIPS CC table header is shown in Appendix B.8.1 on page 111.

The AIPS CC table has no required keywords. The table has either 3 or 4 columns, meaning all components are point sources, or 7 or 8 columns allowing for extended components.

TABLE 67: Mandatory and optional columns for the AIPS CC table

Title	Type	Units	Description
FLUX	1E	Jy	Component total flux
DELTAX	1E	degrees	X separation
DELTAY	1E	degrees	Y separation
DELTAZ	1E	degrees	Z separation
MAJOR AX	1E	degrees	Component major axis
MINOR AX	1E	degrees	Component minor axis
POSANGLE	1E	degrees	Component position angle
TYPE OBJ	1E		Component type code

Component flux. The total flux density of the model component is given in Jy in the FLUX column. It may be of either sign.

Component position. The position of the source relative to the reference position in degrees is given in the DELTAX and DELTAY columns for the x and y offset, respectively. In the absence of rotation, the offset is given by $X - X_0$ in x and $Y - Y_0$ in y where the component is found at pixel (X, Y) in the image and the reference pixel is (X_0, Y_0). These may be related to the celestial coordinates (α, δ) through the non-linear projection coordinates appropriate to the image (usually the -SIN projection). If there is rotation, the offsets are in the rotated coordinates.

The above coordinates are in the frame of the particular facet image. When the DELTAZ column is present, the three offsets represent the 3-vector which is multiplied by the (u, v, w) vector of the uv data set to give the model phase. In other words, the (x, y) in the facet plane has been rotated to the frame of the central facet uv data which causes a z term. Multiple facet CC files are converted to a single "3D" CC file by task MOD3D. These CC models are only suitable for DFT model computation in *AIPS*.

Component dimension. The component major axis, minor axis, and position angle in degrees are specified by the MAJOR AX, MINOR AX, and POSANGLE columns, respectively. For Gaussians, the widths are full width at half maximum. For spheres, the widths are diameters. The position angle is measured counter-clockwise from up (north in the absence of rotation).

Component type. The component type is point if there are fewer than 7 columns in the table and is given be the TYPE OBJ column otherwise. A value of 0 means point, 1, Gaussian, 2, convolved Gaussian, and 3 uniform optically-thin sphere. Gridded modeling routines can handle models of type 0 and 1 if all components in a given AIPS CC table are the same size. DFT modeling routines handle types 0, 1, and 3 with mixed sizes within a given table.

6.2 AIPS CG Clean Gaussian parameters table

In spectral-line imaging, every image plane in the output cube must be scaled so that its units are Jy/beam using the "beam" in the header. However, there is no reason not to use the fits to the dirty beam in each channel as the restoring beam for that channel. Indeed, this is the only way to insure that the Clean component model and the residuals are on approximately the same units. The AIPS CG table is used to record the actual restoring beam as a function of channel for the data cube. It is used to let model fitting routines use the correct restoring beam for each channel and to provide the convolution task CONVL with the information needed to make each spectral channel the same resolution. An example AIPS CG table header is shown in Appendix B.8.2 on page 111.

TABLE 68: Mandatory keywords for AIPS CG table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS CG'
TABREV	I	Table format version number (1)

TABLE 69: Mandatory columns for the AIPS CG table

Title	Type	Units	Description
FREQUENCY	1D	Hz	Channel frequency
BMAJ	1E	degrees	Restoring beam FWHM major axis
BMIN	1E	degrees	Restoring beam FWHM minor axis
BPA	1E	degrees	Restoring beam position angle

Channel frequency. The frequency of the spectral channel to which the other parameters in the row apply is given in the FREQUENCY column in Hz.

Restoring beam. The full width at half maximum of the Gaussian restoring beam is specified in the BMAJ, BMIN, and BPA columns. These give the major axis, minor axis, and position angle of the major axis (East from North), respectively, in degrees.

6.3 AIPS MF model-fit table

The AIPS MF table is used to store the results, uncertainties, and numerous other parameters used and found by Gaussian-fitting tasks, especially SAD. An example AIPS MF table header is shown in Appendix B.8.3 on page 112.

TABLE 70: Mandatory keywords for AIPS MF table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS MF'
REVISION	I	Table format version number (4)
DEPTH1	I	Pixel location on 3rd axis of fit plane
DEPTH2	I	Pixel location on 4th axis of fit plane
DEPTH3	I	Pixel location on 5th axis of fit plane
DEPTH4	I	Pixel location on 6th axis of fit plane
DEPTH5	I	Pixel location on 7th axis of fit plane
REALRMS	I	RMS of image (ignoring signal regions)

Image plane of fit. The plane selected from axes 3 through 7 of the image is given by the values of the DEPTH1 through DEPTH5 keywords, respectively.

Image uncertainty. The "real" (robust, ignoring true signal regions) rms of the image is given by the value of the REALRMS keyword.

Each row of the table shall contain the fit parameters and other information for a single point or Gaussian model component. The units shown in Table 69 are those of a "standard" interferometric image having brightness units of Jy/beam and celestial coordinates for the first two axes. In fact, the units in the table shall match the image planes for which the components are fit. The peak values shall be in the brightness units given in the header and the total values will also be in those units unless they are recognizable as `'JY/'` or `'MAG/'` in which case they become Jy and mag, respectively. The X and Y coordinate values will match those of the X and Y axes, which allows them to be velocity, frequency, or others as well as the standard celestial axes.

TABLE 71: Mandatory columns for the AIPS MF table

Title	Type	Units	Description
PLANE	1E		Plane in image cube
PEAK INT	1E	Jy/beam	Fit peak brightness
I FLUX	1E	Jy	Fit total flux
DELTAX	1E	degrees	Fit X position
DELTAY	1E	degrees	Fit Y position
MAJOR AX	1D	degrees	Fit major axis
MINOR AX	1E	degrees	Fit minor axis
POSANGLE	1E	degrees	Fit position angle
Q FLUX	1E	Jy	Q polarization flux
U FLUX	1E	Jy	U polarization flux
V FLUX	1E	Jy	V polarization flux
ERR PEAK	1E	Jy/beam	Uncertainty in peak brightness
ERR FLUX	1E	Jy	Uncertainty in total flux
ERR DLTX	1E	degrees	Uncertainty in X position
ERR DLTY	1E	degrees	Uncertainty in Y position
ERR MAJA	1E	degrees	Uncertainty in major axis
ERR MINA	1E	degrees	Uncertainty in minor axis
ERR PA	1E	degrees	Uncertainty in position angle
ERR QFLX	1E	Jy	Uncertainty in Q total flux
ERR UFLX	1E	Jy	Uncertainty in U total flux
ERR VFLX	1E	Jy	Uncertainty in V total flux
TYPE MOD	1E		Model type: 0 point, 1 Gaussian
D0 MAJOR	1E	degrees	Deconvolved major axis
D0 MINOR	1E	degrees	Deconvolved minor axis
D0 POSAN	1E	degrees	Deconvolved position angle
D- MAJOR	1E	degrees	Deconvolved major axis at minus 1 sigma
D- MINOR	1E	degrees	Deconvolved minor axis at minus 1 sigma
D- POSAN	1E	degrees	Deconvolved position angle at minus 1 sigma
D+ MAJOR	1E	degrees	Deconvolved major axis at plus 1 sigma
D+ MINOR	1E	degrees	Deconvolved minor axis at plus 1 sigma
D+ POSAN	1E	degrees	Deconvolved position angle at plus 1 sigma
RES RMS	1E	Jy/beam	RMS residual in fit area
RES PEAK	1E	Jy/beam	Peak residual in fit area
RES FLUX	1E	Jy	Total flux in residual in fit area
CENTER X	1E	pixels	Fit X position
CENTER Y	1E	pixels	Fit Y position
MAX AXIS	1E	pixels	Fit major axis
MIN AXIS	1E	pixels	Fit minor axis
PIXEL PA	1E	degrees	Fit position angle in pixel fit
PB FACT	1E		Factor to correct for primary beam
DLY FACT	1E		Factor to correct for bandwidth smearing

Location in cube. The plane number, counting through all axes 3 through 7, shall be given in the `PLANE` column as a positive number of integer value.

Fit Gaussian parameters. The peak intensity, position, and Gaussian widths are given in the `PEAK INT`, `DELTAX`, `DELTAY`, `MAJOR AX`, `MINOR AX`, and `POSANGLE` columns. The total flux computed from them is given in the `I FLUX` column. The uncertainties in these parameters are given in the `ERR PEAK`, `ERR FLUX`, `ERR DLTX`, `ERR DLTY`, `ERR MAJA`, `ERR MINA`, `ERR PA`, and `ERR FLUX` columns, respectively.

Polarization. The `Q FLUX`, `U FLUX`, `VFLUX` columns are reserved for the fit of the component in polarized images, if any. The uncertainties in those fits will appear in the `ERR QFLX`, `ERR UFLX`, and `ERR VFLX` columns, respectively.

Model type. The `TYPE MOD` shall contain the value of the code representing the type of model fit. Codes 0.0 for point and 1.0 for Gaussian are recognized.

Component deconvolution. The fit Gaussian widths may be deconvolved with the Clean beam parameters to obtain an estimate of the true source angular size. The deconvolution is done using the fit widths with the result given in the `DO MAJOR`, `DO MINOR` and `DO POSAN` columns. The deconvolution is also done over all nine possible combinations of the three parameters at 0 and $\pm 1\sigma$. The `D- MAJOR`. `D- MINOR`, and `D- POSAN` columns report the smallest values found, while the `D+ MAJOR`. `D+ MINOR`, and `D+ POSAN` the largest values so found.

`Component fit.` The component fit is initially done in image pixels and then converted to physical parameters. The results of the pixel fit are given in the `CENTER X`, `CENTER Y`, `MAX AXIS`, `MIN AXIS`, and `PIXEL PA` columns.

`Correction factors.` If the user has requested correction for the primary beam, the fit peak and flux values and their uncertainties as reported in the `AIPS MF` table are corrected by the factor reported in the `PB FACT` column. If the user has requested corrections for bandwidth smearing, the peak value and its uncertainty as reported in the `AIPS MF` table are corrected by the factor reported in the `DLY FACT` column.

6.4 `AIPS RM` Rotation measure fitting table

The `AIPS RM` is a large table used by the spectral rotation measure fitting task `RMFIT` It is designed to allow the user to access the fitting over multiple sessions until the result appears complete. At that point, output images of the fit parameters may be generated. An example `AIPS RM` table header is shown in Appendix B.8.4 on page 114.

TABLE 72: Mandatory keywords for `AIPS RM` table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS RM'
YBLC	A	Image cube blc pixel on 2nd axis
YTRC	I	Image cube trc pixel on 2nd axis
ZBLC	I	Image cube blc pixel on 3rd axis
ZTRC	I	Image cube trc pixel on 3rd axis
YINC	I	Pixel increment on 2nd axis in first pass
ZINC	I	Pixel increment on 3rd axis in first pass
ICLIP	D	Minimum spectral value in Ipol for fitting
PCLIP	D	Minimum spectral value in Ppol for fitting
DOSPIX	I	Fitting spectral index or thickness models?
PIX FIT	I	Number pixels already fit

Pixels included in table . Keywords `YBLC`, `YTRC`, `ZBLC`, and `ZTRC` specify which pixels from the image cube appear in the table. `YINC` and `ZINC` are the increments for `RMFIT` to take on those axes while doing the first pass of solutions.

Which spectra to fit. Keyword `ICLIP` specifies the minimum brightness in each spectrum total intensity (I pol.) to allow it to be fit. Keyword `PCLIP` specifies the minimum brightness in total polarization ($\sqrt{(Q^2+U^2)}$) to allow it to be fit. When `RMFIT` restarts, the levels may be reduced.

`Spectral index or thickness fitting?.` The keyword `DOSPIX` contains code values to describe additional spectral fitting besides the rotation measure. Zero means none, 1 means spectral index, 2 means a thickness model ($\sin(x)/x$), 3 Gaussian thickness model, and 4 exponential thickness mode.

`Number of pixels already fit.` Keyword `PIX FIT` is used to document the number of pixels which already have Gaussian-fit results.

TABLE 73: Mandatory columns for the `AIPS RM` table

Title	Type	Units	Description
`PIXEL`	2I		Y, Z pixel position in the cube
`NCOMP`	1I		Number components actually fit
`IFLUX`	1E	*B_units*	Peak Ipol spectral value
`PFLUX`	1E	*B_units*	Peak Ppol spectral value
`PONE`	4E	*B_units*	Fit polarization at 1 GHz
`THETA`	4E	degreees	Fit polarization angle at 0 GHz
`ROT MEAS`	4E	radians/m^2	Fit rotation measure
`SPECINDX`	4E		Fit spectral index
`ERR P1`	4E	*B_units*	Uncertainty in Pone
`ERR THET`	4E	degrees	Uncertainty in fit polarization angle
`ERR RM`	4E	radians/m^2	Uncertainty in fit rotation measure
`ERR SPIX`	4E		Uncertainty in fit spectral index
`THICKNESS`	4E	radians/m^2	Fit thickness
`ERR THICK`	4E	radians/m^2	Uncertainty in fit thickness

Table initial values. The `PIXEL` column contains the Y, Z pixel position from the input cube and the `IPEAK` ans `PPEAK` columns contain the peak values in the Ipol and Ppol spectra, respectively, at that pixel. All other columns are filled with zeros or magic blanks.

Fit results. The remaining columns contain the results of fitting the rotation measure model to the spectrum at that pixel. The fit total polarization (at 1 GHz is spectral index or thickness are computed) is stored in the `PONE` column in brightness units, the polarization intrinsic angle is stored in the `THETA` column, the rotation measure in radians per meter squared is stored in the `ROT MEAS` column, and the spectral index parameter is stored in the `SPECINDX` column. When thickness is fit, the spectral index columns are replaced with a `THICKNESS` column. The uncertainties in these parameters are stored in the corresponding `ERR` *xxxx* columns.

The model being fit is

$$
\begin{aligned}
Q(i) &= \sum P_j \cos(2\theta_j + 2R_j\lambda(i)^2)F(\beta_j, , \lambda(i)^2) \\
U(i) &= \sum P_j \sin(2\theta_j + 2R_j\lambda(i)^2)F(\beta_j, , \lambda(i)^2)
\end{aligned}
$$

where P_j is the polarized flux of component j, θ_j is the polarization angle of component j, R_j is the rotation measure of component j, $\lambda(i)$ is the wavelength of spectral channel i, β_j is a "thickness" parameter and F is one of

$$
\begin{aligned}
F(\beta_j, \lambda(i)^2) &= \left(\frac{\lambda_1^2}{\lambda(i)^2}\right)^{\frac{\beta_j}{2}} \\
&= \frac{\sin(\beta_j\lambda(i)^2)}{\beta_j\lambda(i)^2} \\
&= e^{-\ln(2)\left(\frac{\beta_j\lambda(i)^2}{1.8954}\right)^2}
\end{aligned}
$$

$$= \quad e^{-\ln(2)\frac{\beta_j \lambda(i)^2}{1.8954}}$$

where λ_1 is the wavelength at the fiducial frequency of 1 GHz and the thickness models will be referred to by the names spectral index, slab, Gaussian, and exponential, respectively. The peculiar number 1.8954 was chosen so that the last three models reach the 0.5 point at the same value of β_j.

6.5 AIPS ST stars table

The AIPS ST table is a list of fiducial points in the present image. These may represent anything the user wishes, but have been used to mark stars, model-fit Gaussians, and special points within the image. An example AIPS ST table header is shown in Appendix B.8.5 on page 115.

There are no required keywords in the AIPS ST table. This table is unusual in that its first two columns are labeled with, and have the units of, the first two axes of the image. Each row of the table contains the parameters of a different "star."

TABLE 74: Mandatory columns for the AIPS ST table

Title	Type	Units	Description
X_type	1D	*X_units*	Star X location
Y_type	1D	*Y_units*	Star Y location
MAJOR AX	1E	*X_units*	Star major axis
MINOR AX	1E	*Y_units*	Star minor axis
POSANG	1E	degrees	Star position angle
STARTYPE	1E		Type code
LABEL	24A		String with which to label star

Star location. The location columns are named with the strings found in the first two axis types of the image header. They are supplied with suitable units which are degrees for celestial axes, seconds for time, Hz for frequency, m/sec for velocities, and pixels for blank.

Star extent. The star may be considered to have an extent in the X and Y axes with position angle 0.0 for two unlike axes. For celestial axis pairs, the extent becomes a true major axis, minor axis, and position angle (East from North to the major axis). These are given in the MAJOR AX, MINOR AX, and POSANG columns, respectively.

Star type. The type of star may be specified by a code which shall be a positive integer and specified in the STARTYPE column. *AIPS* recognizes values 1.0 through 24.0 and plots each with a different symbol. Values ≤ 0.0 are taken to mean no type and, hence, to *AIPS* no plotted symbol.

Star label. An arbitrary 24-character string, blank filled after the last printable character, is included in the LABEL column. *AIPS* offers the option of plotting the string next to the star symbol.

6.6 AIPS XG Gaussian fitting table

The AIPS XG is a large table used by the spectral Gaussian fitting tasks XGAUS and AGAUS It is designed to allow the user to access the fitting over multiple sessions until the result appears complete. At that point, output images of the fit parameters may be generated. An example AIPS XG table header is shown in Appendix B.8.6 on page 116.

TABLE 75: Mandatory keywords for AIPS XG table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS XG'
YBLC	A	Image cube blc pixel on 2nd axis
YTRC	I	Image cube trc pixel on 2nd axis
ZBLC	I	Image cube blc pixel on 3rd axis
ZTRC	I	Image cube trc pixel on 3rd axis
YINC	I	Pixel increment on 2nd axis in first pass
ZINC	I	Pixel increment on 3rd axis in first pass
VCLIP	D	Minimum spectral value for fitting
VOFF	D	Central value on 1st (spectral) axis
REFVALUE	D	Reference value for spectral axis
REFPIXEL	D	Reference pixel for spectral axis
REFINCR	D	Reference increment for spectral axis
REFTYPE	C	Type of spectral axis
ABSORPTN	I	1 -¿ absorption fit, else emission
PIX FIT	I	Number pixels already fit

Pixels included in table . Keywords YBLC, YTRC, ZBLC, and ZTRC specify which pixels from the image cube appear in the table. YINC and ZINC are the increments for XGAUS to take on those axes while doing the first pass of solutions.

Which spectra to fit. Keyword VCLIP specifies the minimum brightness in each spectrum to allow it to be fit. When XGAUS restarts, the level may be reduced.

Old reference value on 1st axis. Keyword VOFF specifies the spectral (1st) axis reference value when the fitting is commenced. It is used to confirm that restarts of the task are meaningful.

Spectral axis values and type. Keywords REFVALUE, REFPIXEL, REFINCR, and REFTYPE are used to define the coordinate values and type on the spectral axis.

Absorption versus emission. The mathematics of fitting emission Gaussians (XGAUS) or absorption Gaussians (AGAUS) are slightly different. Value 0 implies emission, 1 implies absorption.

Number of pixels already fit. Keyword PIX FIT is used to document the number of pixels which already have Gaussian-fit results.

Table initial values. The PIXEL column contains the Y, Z pixel position from the input cube and the PEAK column contains the peak value in the spectrum at that pixel. All other columns are filled with zeros or magic blanks.

Fit results. The remaining columns contain the results of fitting Gaussians to the spectrum at that pixel. The baseline and slope, if fit, are entered in brightness units and brightness units per X pixel. The fits for up to 8 Gaussians are entered in the AMPLITUDE, CENTER, and WIDTH columns and their uncertainties are entered in the corresponding ERR columns. Units are image brightness units and pixels. The number of Gaussians actually fit at the pixel is entered in the NGAUS column.

TABLE 76: Mandatory columns for the AIPS XG table

Title	Type	Units	Description
PIXEL	2I		Y, Z pixel position in the cube
NGAUS	1I		Number Gaussians actually fit
PEAK	1E	*B_units*	Peak spectral value
BASELINE	2E	*B_units*	Fit baseline, slope
ERR BASE	2E	*B_units*	Uncertainty in fit baseline
AMPLITUD	8E	*B_units*	Fit amplitude
CENTER	8E	pixels	Fit center
WIDTH	8E	pixels	Fit width
ERR AMPL	8E	*B_units*	Uncertainty in fit amplitude
ERR CENT	8E	pixels	Uncertainty in fit center
ERR WIDT	8E	pixels	Uncertainty in fit width

6.7 AIPS ZE Zeeman splitting fitting table

The AIPS ZE is a large table used by the spectral Zeeman splitting fitting tasks ZEMAN and ZAMAN It is designed to allow the user to access the fitting over multiple sessions until the result appears complete. At that point, output images of the fit parameters may be generated. An example AIPS ZE table header is shown in Appendix B.8.7 on page 117.

TABLE 77: Mandatory keywords for AIPS ZE table headers

Keyword	Value type	Value
EXTNAME	A	'AIPS ZE'
YBLC	A	Image cube blc pixel on 2nd axis
YTRC	I	Image cube trc pixel on 2nd axis
ZBLC	I	Image cube blc pixel on 3rd axis
ZTRC	I	Image cube trc pixel on 3rd axis
DOGAUS	I	Type of Zeeman fit
ICLIP	D	Minimum spectral value in I pol. for fitting
REFVALUE	D	Reference value for spectral axis
REFPIXEL	D	Reference pixel for spectral axis
REFINCR	D	Reference increment for spectral axis
REFTYPE	C	Type of spectral axis
ABSORPTN	I	1 -¿ absorption fit, else emission
PIX FIT	I	Number pixels already fit

Pixels included in table . Keywords YBLC, YTRC, ZBLC, and ZTRC specify which pixels from the image cube appear in the table.

How fitting is done. Keyword DOGAUS specifies the style of Zeeman splitting: -1 uses a 2-sided derivative method, 0 uses a 1-sided derivative, and $+1$ uses the Gaussians found previously by XGAUS.

Which spectra to fit. Keyword ICLIP specifies the minimum brightness in each I polarization spectrum to allow it to be fit. When ZEMAN restarts, the level may be reduced.

Spectral axis values and type. Keywords REFVALUE, REFPIXEL, REFINCR, and REFTYPE are used to define the coordinate values and type on the spectral axis.

Absorption versus emission. The mathematics of fitting emission Gaussians (XGAUS) or absorption Gaussians (AGAUS) are slightly different. Value 0 implies emission, 1 implies absorption.

Number of pixels already fit. Keyword PIX FIT is used to document the number of pixels which already have Gaussian-fit results.

TABLE 78: Mandatory columns for the AIPS ZE table

Title	Type	Units	Description
PIXEL	2I		Y, Z pixel position in the cube
I PEAK	1E	*B_units*	Peak Ipol spectral value
GAIN	1E		Fit Gain
ERR GAIN	1E		Uncertainty in fit Gain
B FIELD	8E	pixels	Fit line splitting
ERR BFLD	8E	pixels	Uncertainty in fit line splitting
NGAUS	1I		Number Gaussians actually fit
AMPLITUD	8E	*B_units*	Fit amplitude
CENTER	8E	pixels	Fit center
WIDTH	8E	pixels	Fit width
BASELINE	2E	*B_units*	Fit baseline, slope

Table initial values. The PIXEL column contains the Y, Z pixel position from the input cube and the I PEAK column contains the peak I polarization value in the spectrum at that pixel. The GAIN and B FIELD columns are their uncertainties are filled with magic blanks. If the user has requested the Gaussian method of Zeeman fitting, the NGAUS, AMPLITUD, CENTER, WIDTH, and BASELINE columns are filled with values taken from the I polarization XG table.

Fit results. The Zeeman model that is fit to the V polarization is a gain times the I polarization A "B field" times the slope if the I polarization with channel. The results are stored in the GAIN and B FIELD columns with the unceratinties stored in the ERR GAIN and ERR BFLD columns. Note that the "B field" is in X axis pixels; any association of that separation with an actual magnetic field is up to the user.

7 Revision history

TABLE 79: Substantive revisions

Date	Type	Content
2020-03-04	CC	Add DELTAZ optional column, changing meaning of DELTAX and DELTAY columns See page 57.
2019-10-14	RM	Newly documented table for rotation measure fitting of image cubes. See page 60.
2019-10-11	PP	New table for right-left phase difference correction spectrum. See page 53.
2019-10-11	XG	Newly documented table for Gaussian fitting of image cubes. See page 63.
2019-10-11	ZE	Newly documented table for Zeeman splitting fitting of image cubes. See page 64.
2019-10-11	PO	Change keyword SOURCE ID to SOURCE_ID. See page 35.
2017-06-19	CD	Addition of SOLCAL1 and SOLCAL2 columns. See page 15.
2017-06-19	SY	Addition of CAL TYPE column. See page 37.
2016-02-16	WX	Addition of WIND_GUST and PRECIPITATION columns. See page 40.
2015-04-24	SN	Addition of DISP and DDISP columns. See page 54.
2015-04-24	UV	Addition of SUBARRAY, ANTENNA1, and ANTENNA2 alternatives to BASELINE. See page 8–9.
2014-03-24	CC	Description of DELTAX and DELTAY corrected. See page 57.
2012-08-28	SU	Remark about when SU required, when normally omitted. See page 35.
2012-06-28	UV-table	SOURCE and FREQSEL columns are mandatory only under certain conditions. See pages 11 and 11.
2012-05-30	AN	Add comments about the x, y coordinate systems used for the array center and station coordinates. See pages 12 and 14.
2012-05-25	FQ	Add BANDCODE column. See pages 23 and 23.
2012-02-21	SU	Add RAOBS and DECOBS columns. See pages 36 and 37.

A Summary of the FITS format

A.1 Introduction and history

Astronomy is alone among the sciences in having an international standard data interchange format that is used by virtually all scientists and institutions in the field. This format is named FITS or Flexible Image Transport System although by now *Information* would be a better word than merely *Image*.[1] FITS is primarily a syntactic rather than semantic standard allowing the unambiguous interchange of descriptive text and binary image and tabular data. The meaning to be associated with the various fields of the data remains obscure in many cases. Enormous effort has gone into defining some of the needed semantics so that, at least, coordinate information may be transmitted meaningfully along with the data. This appendix will begin with an outline of the history of the development of FITS and will then present a few of the technical details of the format. If the reader is required to write software to read and/or write FITS-format data, he or she will need to consult the latest codification of the FITS format (Pence *et al.* 2010 [21]) and will probably wish to use the `CFITSIO` software package (Pence 1992 [19], 1999[20]). Both of these plus copies of all FITS papers and much more may be found from the web site maintained at the Goddard Space Flight Center `http://fits.gsfc.nasa.gov/`.

In the mid 1970s, programmers from some of the major data producing centers realized the need to exchange, for example, radio and optical images of the same field. They began informal attempts at this exchange, which remained largely unknown to the rest of the community. These programmers were reacting to the increasingly burdensome task of writing format translation programs to convert data from the various instruments at their institutions to the multiple software packages at their institutions. The number of translation programs between software packages grows as the square of the number of packages while the probability that these translation routines are up-to-date with the continually changing internal formats diminishes rapidly. Inclusion of data from another institution — a significant requirement scientifically — would be almost unsupportable.

With encouragement from the U. S. National Science Foundation, a representative from the Kitt Peak National Observatory (KPNO, now National Optical Astronomy Observatory) met with a representative of the National Radio Astronomy Observatory (NRAO) at the site of the NRAO Very Large Array Telescope March 27 and 28, 1979. The meeting was also attended by a number of NRAO's scientific staff, most notably Barry Clark. Remarkably, the basic FITS format was designed by Don Wells (KPNO) and Eric Greisen (NRAO) and described in writing in these two days. In April 1979, the first magnetic tapes to use the new format were interchanged between the two institutions using possibly the worst possible combination of software and hardware environments. The first FITS files were written by a PL/I program on an IBM 360 under OS/MFT (32-bit, twos-complement numbers and 8-bit EBCDIC characters) and were read by a Fortran program executing on a CDC 6400 under SCOPE (60-bit, ones-complement numbers and 6-bit "Display Code" characters). Despite these obstacles, the images were transmitted without error on the first try. Those 30-year old files remain legitimate FITS files to this day and may still be read and understood by any conforming FITS reading software.

The FITS format contains its data in what is now known as a "header-data unit" (HDU). An HDU begins with descriptive information in human-readable form, the header, and is then followed by the data in binary form. The initial FITS paper described the transmission of n-dimensional arrays of data ("images"), but was wise enough to allow "special records" of any sort to follow the described HDU, so long as the logical record length of the special records was the 2880-bytes required of the image HDU. In October 1979, Greisen and Ron Harten (Netherlands Foundation for Radio Astronomy) took advantage of this escape clause to define a similar HDU more suitable to data from radio astronomical interferometers. This "random groups" HDU described sets of n-dimensional arrays, each accompanied by a number of binary "random parameters" describing the array. A set of small images of the sky, each accompanied by random parameters describing the coordinates of the image would be a simple example. The random-groups format was not widely used except by radio interferometrists and is now usually replaced by the binary table format to be discussed

[1] FITS has been adopted as the archival format for a project to digitize the entire Vatican Library — 80,000 manuscripts expected to require 45 petabytes of storage! It is also used in nuclear medicine among other fields.

below.

The basic FITS agreement was published by Wells, Greisen, and Harten in 1981 [25] and the Greisen and Harten [11] extension paper appeared in the same issue of *Astronomy & Astrophysics Supplement Series*. By that time, FITS had already become the *de facto* interchange format for astronomy. Recognizing this fact, the Chairman and Co-Chairman of Commission 5 of the IAU, Bernard Hauck and Gart Westerhout, asked this author to recommend a resolution for Commission 5 at the 1982 meeting in Patras, Greece. It was adopted (IAU, 1982[16]) and a working group to develop further extensions to FITS was established eventually (IAU 1988 [17]) under the leadership of Preben Grosbøl.

Numerous FITS extensions, known only to their inventors, followed. To provide guidelines for defining *conforming* extensions, Grosbøl *et al.* negotiated in 1984 and finally published in 1988 [12] a general HDU structure with a few new required keywords to be used in future conforming extensions. Like the original agreement, this new agreement allowed any number of conforming extension HDUs to follow the primary HDU and their association was to be inferred by their presence in the same tape or disc file. The authors enunciated a principle that "The most important rule for designing new extensions to FITS is that existing FITS tapes must remain valid." This "once FITS, always FITS" principle is one of the reasons for the success of the whole FITS experience.

Papers describing specific extensions in this new scheme followed. Harten *et al.* (1988) [15] described an extension to contain tabular data fully in printable ASCII character form. Ponz *et al.* (1994) [23] then described a conforming extension to hold an image which might have different dimensionality, coordinates, and binary type from the primary image (if any). This allowed multiple associated images to be contained in a single file in multiple instances of this "image" extension. Cotton *et al.* (1995) [3] defined a "binary-tables extension" which was prototyped as early as 1984, negotiated into a general extension in 1991, and finally published in 1995. This extension conveys tabular data in a more efficient binary form and allows each column to be in its own, most suitable, binary type. A column may contain an array of data of any defined size, but each row must be the same length in bytes. This extension thereby encompasses all of the previous data forms with the only differences being in the header keywords of the HDU.

Wells *et al.* (1981) [25] recognized the need for world coordinate system (WCS) keywords and provided keywords for each axis of the image to specify coordinate type and a reference point for which the pixel coordinate, a coordinate value, and an increment were given. This description was deliberately kept simple to avoid controversy and was not universally adopted. A NASA-sponsored conference in 1988 recommended the development of a World Coordinate System standard for use within FITS (Hanisch & Wells 1988 [14]) based on the one already in use (Greisen 1983 [4], 1986 [5]). The negotiations on this point have been the most difficult and protracted of any FITS negotiations and continue to this day. The GSFC web site contains reference to many of the intermediate presentations on this matter. Finally, Greisen & Calabretta (2002 [9]) published paper I of the WCS agreement describing a general form by which the various coordinate types would be described. Paper II describing celestial coordinates appeared at the same time written by Calabretta & Greisen (2002 [1]). Paper III on spectral coordinates appeared three years later (Greisen, Calabretta, Valdes, & Allen, 2005 [10]). A paper on time coordinates is well advanced at this writing, while the paper on corrections for instrumental distortions appears to be stalled.

The papers described above have had to be written in a style not entirely suited to a description of a standard. Because they presented new ideas, they had to be filled with explanations and justifications for the various choices made, as well as examples of the uses of the format and of the coordinates. To re-write these papers into a more correct standards definition, a number of panels have been convened with the support of NASA. The results have been a number of documents which have been discussed and adopted by the IAU FITS Working Group. Version 2.0 of this standard was published (Hanisch *et al.* 2001 [13]) and version 3.0 (Pence *et al.* 2010 [21]) has now also been published. All versions may be found at the GSFC web site. It is this document which should be studied by authors of FITS reading and writing software.

Further discussion of the history of the FITS development may be found in Wells (2000 [24]) and Greisen (2003 [6] and [7]).

Table 1: Primary header required keywords

Keyword	=	**Value**	**Comment**
`SIMPLE`	=	T	/ Identifies a FITS file
`BITPIX`	=	b	/ Binary data encoding
`NAXIS`	=	N	/ Number axes in array
`NAXIS1`	=	n_1	/ Number pixels fastest changing axis
`NAXIS2`	=	n_2	/ Number pixels 2nd fastest changing axis
...			/ `NAXIS`j as needed
`NAXIS`N	=	n_N	/ Number pixels slowest changing axis
...			other keywords
`END`			

A.2 Basic FITS

In the modern world, data are kept in files on disc which are simply byte streams. However, in the world of 1979, data files were frequently kept on external magnetic media such as 9-track tapes. For such media, the logical and physical record lengths are quite significant. FITS adopted a logical record length of 2880 8-bit bytes for all records in a FITS file. This curious length happens to be an integer number of words on all computers ever manufactured, simplifying the reading and writing of FITS records. It is long enough to be reasonably efficient on 9-track tapes and short enough to avoid strain on the small-memory computers of that day.

All "conforming" HDUs in FITS have the same general structure. The HDU begins with one or more 2880-byte logical records in ASCII characters, each structured as 36 80-character "card images." Each such card image begins with an 8-character upper-case ASCII keyword usually followed by a blank character and an equals sign. A value to be assigned to the keyword follows, optionally followed by a "/" character and commentary information. A modest number of keywords are required and must come at the beginning of the header in a specified order. They identify the nature of the HDU and define the binary format and dimensions of the data portion. Other keywords which describe how to decode the data may also be required, but may come in any order. The last keyword in each header is `END` and the last logical record in the header is filled with ASCII blanks.

The data portion, if any, of each HDU begins in the first logical record following the header. The data are a fully-packed byte stream broken into 2880-byte logical records with no padding, except that the last logical record is filled out as needed with binary zeros. Following the data portion of the HDU is either an end-of-file, another conforming HDU, or "special records" of length 2880-bytes and no formally defined structure.

The initial HDU defined by Wells, Greisen, & Harten (1981 [25]) described an n-dimensional, regularly-spaced array. The required keywords are shown in the required order in Table 1. The `SIMPLE` keyword has a logical value `T` and is the "signature" of a FITS file. `BITPIX` has one of six integer values whose meanings are listed in Table 2. `NAXIS` has an integer value from 0 to 999 giving the number of axes in the N-dimensional array. It is followed immediately by `NAXIS`j keywords giving the number of pixels on each of the N axes. Any number of other keywords may follow. They may be taken from the standard lists of defined keywords (so long as they are used for the defined purpose) or may be any undefined keyword of the user's choosing. The definition document (Pence *et al.* 2010 [21]) describes the use of the defined keywords for such things as dates, telescope, source name, history, comment, coordinates, and the like. The `END` keyword is followed by ASCII blanks to the end of the logical record in which it occurs.

Table 2: Valid values of keyword BITPIX

Value	**Binary data representation**
8	Character or unsigned binary integer
16	16-bit two's complement binary integer
32	32-bit two's complement binary integer
64	64-bit two's complement binary integer
-32	IEEE single-precision floating-point
-64	IEEE double-precision floating-point

The binary data begin in the first logical record following the header and are in the binary format specified by the value of keyword BITPIX, where, for image data, the value 8 means unsigned, 8-bit binary integers (0 – 255). Using the absolute value of BITPIX, the total number of bits occupied by the binary data is

$$N_{\rm bits} = |\texttt{BITPIX}| \times (\texttt{NAXIS1} \times \texttt{NAXIS2} \times \ldots \times \texttt{NAXIS}N)\,, \tag{5}$$

where $N_{\rm bits}$ is non-negative. The binary numbers are in the byte order normal for "big-endian" computers. The byte that contains the sign bit is first and the byte that contains the ones bit is last. (Computers based on Intel and AMD cpu chips are "little endian," meaning that FITS readers and writers on those computers must swap the byte order between internal and FITS representations of the binary numbers.) Following the last byte of the $N_{\rm bits}$ of data, the remainder of the logical record is filled with zeros.

Note that the definitions above allow there to be no data array whatsoever. The random-groups extension to the basic image FITS was the first extension to employ this loophole. It is an IAU endorsed format widely used for radio interferometric data. See Greisen & Harten (1981 [11]) or the current FITS Definition (Pence, *et al.* 2010) for details. Since the future use of random groups has been "deprecated" and fully replaced by the binary table extension, we will not describe it here. Nonetheless, many valid FITS data files are written in which all significant, non-header information is contained within the extension HDUs.

A.3 Conforming FITS extensions

Grosbøl *et al.* (1988 [12]) defined a standard to be used by all conforming extensions to FITS. The required keywords of all extension headers are shown in Table 3 in the required order. The XTENSION keyword gives a character-valued name for the type of extension; values TABLE, IMAGE, and BINTABLE have since been defined and endorsed by the IAU FITS Working Group. The PCOUNT keyword defines the number of extra data values that accompany each data structure while GCOUNT defines the number of data structures. These two keywords are inherited from the random-groups format and have specific values and meanings in each of the three extensions so far defined. The total number of bits in the binary data of a conforming extension is given by

$$N_{\rm bits} = |\texttt{BITPIX}| \times \texttt{GCOUNT} \times (\texttt{PCOUNT} + \texttt{NAXIS1} \times \texttt{NAXIS2} \times \ldots \times \texttt{NAXIS}N)\,, \tag{6}$$

where $N_{\rm bits}$ must be non-negative. The last data record in the HDU is filled with binary zeros or ASCII blanks in the case of the TABLE extension.

The first defined extension was described by Harten *et al.* (1988 [15]). This TABLE extension transmits tabular data in the form of columns and rows all represented in ASCII characters. For this extension, NAXIS is 2, BITPIX is 8, NAXIS1 is the total number of bytes (characters) in a row, NAXIS2 is the number of rows, PCOUNT is zero, and GCOUNT is one. New required keywords define the number of logical columns in the table, the start position in the row for each logical column, and the format used to encode the value of each logical column. Optional reserved keywords may be used to name the columns and their units, to rescale the values in the columns, to define the representation of null (not-valid) values in each column, and to recommend display formats for each column. This extension is useful for transmitting tables of data traditionally maintained in

Table 3: Extension header required keywords

Keyword	=	Value	Comment
XTENSION	=	'*name*'	/ Names extension type
BITPIX	=	b	/ Binary data encoding
NAXIS	=	N	/ Number axes in array
NAXIS1	=	n_1	/ Number pixels fastest changing axis
NAXIS2	=	n_2	/ Number pixels 2$^{\text{nd}}$ fastest changing axis
...			/ NAXISj as needed
NAXISN	=	n_N	/ Number pixels slowest changing axis
PCOUNT	=	p	/ Number parameter values per group
GCOUNT	=	g	/ Number data groups
...			other keywords
END			

character form such as star catalogues. The encoding and decoding of binary data into ASCII is expensive in computer time and normally causes either a loss in accuracy or a substantial expansion in the data volume. Therefore, the binary tables extension, to be discussed below, is now the table extension of choice.

The next extension was described by Ponz *et al.* (1994 [23]) to allow a collection of associated images to be transmitted in a single FITS file. Each IMAGE extension transmits a single n-dimensional array whose format, units, dimensions, coordinates, etc. need not match those of the primary image (if there is one) or those of other IMAGE extensions within the FITS file. The IMAGE extension has been used to transmit images from separate quadrants of a detector device, each of which have different coordinates and image defects, and to transmit aligned images of different physical parameters including images of data reliability. The BITPIX, NAXIS, and NAXISj keywords are used as in basic FITS images, PCOUNT is zero, and GCOUNT is one. No reserved keywords are defined specifically for this extension.

The binary table extension was proto-typed as early as 1984 and was finally published in a more general form by Cotton *et al.* (1995 [3]). This extension allows the transmission of any form of data that may be viewed in a general way as a collection of rows and columns. Each row must have the same length in bytes and all values within a column must have the same binary format and dimensionality. However, each column may have any one of the standard formats including logical, bit, unsigned byte, 16-, 32-, and 64-bit two's complement integer, single- and double-precision floating-point, single- and double-precision complex, and 32- and 64-bit pointers into an additional data area. The value contained within a column may be a scalar or an array. The required and important optional keywords reserved by the BINTABLE extension are listed in Table 4. Two special keywords TFIELDS and TFORMn for $n = 1,$TFIELDS are required, while a number of others are defined and very useful. PCOUNT is used to introduce a completely unstructured data area called the "heap" which follows the well-structured binary table. This data area is a special concept that allows the table column to contain two numbers, a length and a pointer into the heap data area. The effect is to allow variable length arrays within one or more columns of the table. See the FITS Definition at http://fits.gsfc.nasa.gov/ for a detailed description including some advice regarding the use of the heap or variable-length array concept.

The IAU FITS Working Group was set up by Commission 5 of the IAU in 1988 and has been chaired very ably by Preben Grosbøl followed by Donald Wells and now William Pence. This Working Group maintains a registry of conventions that have been layered upon the formally adopted FITS formats. See http://fits.gsfc.nasa.gov/ for details. These include conventions for maintaining a binary checksum of the FITS file, for associating keyword values in a hierarchical structure, for continuing keyword values on more than one 80-byte field, for inheriting primary-header keyword values in a secondary HDU, and for associating HDUs in a hierarchical structure. The conventions also deal with special semantic definitions for specialized areas of astronomy including optical interferometry, multi-beam radio telescopes, and radio

Table 4: Binary table extension header required keywords

Keyword	=	**Value**	**Comment**
`XTENSION`	=	`'BINTABLE'`	/ Names extension type
`BITPIX`	=	8	/ Binary data encoding
`NAXIS`	=	2	/ Number axes in array
`NAXIS1`	=	n_1	/ Number bytes in each row
`NAXIS2`	=	n_2	/ Number rows
`PCOUNT`	=	p	/ Number bytes of heap data
`GCOUNT`	=	1	/ Number data groups
`TFIELDS`	=	C	/ number of columns
`TFORM1`	=	'$format_1$'	/ column 1 format, count
`TFORM2`	=	'$format_2$'	/ column 2 format, count
...			/ other `TFORM`n as needed
`TFORM`C	=	$format_C$'	/ column C format, count
`TTYPE`n	=	'*col_name*'	/ column n name
`TUNIT`n	=	'*col_unit*'	/ column n units
`TSCAL`n	=	S_n	/ column n scaling parameter
`TZERO`n	=	Z_n	/ column n offset parameter
`TDIM`n	=	'$(i, j, k, l, \ldots)$'	/ column n array dimensions
...			other keywords
`END`			

interferometers, particularly very long baseline arrays. This last (Greisen 2011 [8]) provides a detailed example of the extensive implementation of binary tables.

A.4 World coordinates

Almost all of the format described above is syntactic rather than semantic. A fully general FITS reader will correctly interpret all keywords and binary data sent to it, creating the local equivalent of image arrays and tables. But without further agreements on semantics, the actual meaning of the arrays and table columns will not be understood. For data from one telescope to be compared with data from another — and that is much of what modern astronomy is about — the celestial positions, frequencies, polarizations, times, *et al.* of each data value must be known. The transmission of this information has become known as the World Coordinate System (WCS). The methods used to describe the WCS keywords in FITS are summarized in the definition document (Pence *et al.* 2010 [21]) and were initially published by Greisen & Calabretta (2002 [9]), Calabretta & Greisen (2002 [1]), and Greisen *et al.* (2005 [10]) to describe, respectively, the fundamental methods for handling WCS and then apply them to celestial and spectral coordinates.

The simplest method to convey the coordinate values for each value of the data would be to transmit a complete array of coordinates with each value. This would be rather expensive (*e.g.*, five double-precision numbers for each single-precision value) and would not recognize the connections between the coordinates of each value. For example, each two-dimensional image might well have been recorded at a single frequency, polarization, and time with a well-established geometric connection between the celestial coordinates of each pixel. The FITS WCS conventions thus provide for methods to describe the relationship of physical coordinates to pixel position within the data array. For the few remaining pathological cases, a table look-up method has also been defined (see Section A.4.4). The most obvious case requiring table look-up would be a collection of images taken at the same physical coordinates at a set of times determined by the vagaries of weather and time allotment committees.

In the basic view, pixels (p_j) are numbered from 1 to `NAXIS`j along axis j. A linear matrix transformation

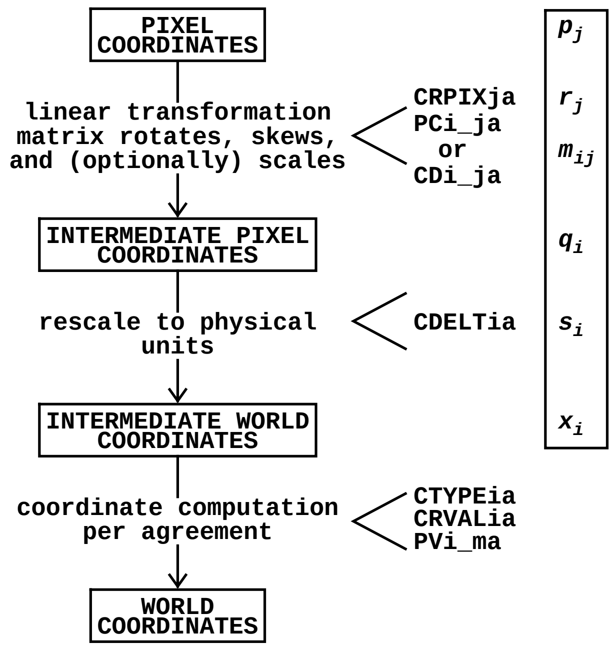


Figure 1: Schematic process for computing world coordinates from pixel coordinates.

(m_{ij} = PCi_j) is applied to the pixel numbers after they have been shifted by a reference pixel (r_j = CRPIXj). In one of the two allowed representations, the "intermediate pixel coordinates" are then scaled to "intermediate world coordinates" (s_i = CDELTi) in physical units (degrees, Hz, etc.), while the other representation combines the scaling into the matrix multiplication. This scheme is illustrated in Fig. 1 and is expressed as

$$x_i = \sum_j s_i m_{ij} (p_j - r_j) \, . \tag{7}$$

The intermediate world coordinates are described as having a value in world coordinates at the reference pixel (CRVALi) and a type (CTYPEi). The 8-character type string takes the form ABCD-XYZ where the ABCD gives the name of the physical coordinate (*e.g.*, FREQ, TIME, RA--, GLAT) while the XYZ gives a label for the possibly non-linear algorithm required to go from intermediate to actual world coordinates. The allowed values for ABCD and XYZ must be established by agreement as must any additional parameters (PVi_m and PSi_m) required by the particular algorithm. Coordinate types not in "4-3" form are taken to be linear. In that case, the world coordinate is the sum of the intermediate world coordinate and the reference value.

Three subtleties should be mentioned at this point, before discussing the details of the prescriptions for particular kinds of coordinates. First, the initial FITS paper by Wells *et al.* (1981) contained an example in which a two-dimensional image was described with NAXIS = 4 in order to attach four physical coordinates to the image. With NAXIS3 = 1 and NAXIS4 = 1 this was a reasonable way to describe the image. However, some software systems were reluctant to accept data with "four" axes, so a new keyword WCSAXES was invented (Greisen & Calabretta, 2002 [9]). This allows, for example, NAXIS = 2 and WCSAXES = 4 to satisfy both the need for four coordinate types with the desire to count only "real" axes. Second, for some data sets, particularly spectroscopic ones, it may be desirable to provide more than one coordinate description. The letter *a* trailing the keywords shown in Fig. 1 is a single character appended to the normal keyword to specify an alternate description. The primary description uses a blank or null character, while secondary descriptions may use any of the letters A through Z. Note that all coordinate keywords must be specified for an alternate description even if they do not differ from those of the primary description. Third, the keywords needed to provide coordinate descriptions for data in tables are defined to be very similar to those appearing in primary and image HDUs. However, since table keywords must contain additional parameters (column

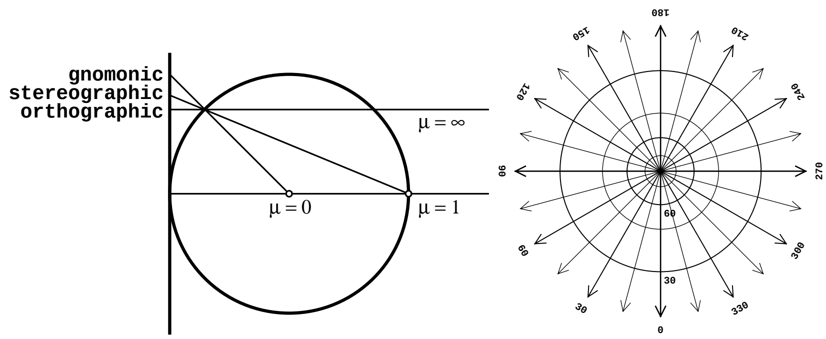


Figure 2: Zenithal projections: (left) geometry showing the three important special cases: gnomic (`TAN`) typical of optical telescope systems, stereographic (`STG`), and orthographic (`SIN`) used by radio interferometers; (right) Gnomic (`TAN`) projection native coordinates, diverges at $\theta = 0$.

number) and cannot be longer than eight characters, the spellings of these keywords are somewhat different. See all of the fundamental WCS papers for discussions of table coordinate keywords and their use.

There are several packages of subroutines to assist you in programming with World Coordinates. The most widely used of these is `wcslib` by Mark Calabretta and all of them may be obtained following links from the main FITS web site (`http://fits.gsfc.nasa.gov/`).

A.4.1 Celestial coordinates

The intermediate world coordinates, (x, y), of an image of the sky need to be converted to longitude and latitude in some celestial system so that images taken by one telescope may be compared with those taken with a different telescope. There are a wide variety of projections of the celestial sphere onto a plane, some of which may be seen as literal optical projections with various locations for the light source while others are cylindrical or conic projections and still others are mathematical constructs allowing the full celestial sphere to be imaged. Many have special properties, such as equal area, which suggest their use for particular imaging problems. Calabretta & Greisen (2002 [1]) describe 26 of these projections in mathematical detail while Calabretta & Roukema (2007 [2]) have added the HEALpix projection used to aid analysis of data over the full sphere. The intermediate world coordinates (or projection-plane coordinates) are first converted to "native spherical coordinates", (ϕ, θ), on a coordinate system determined by the type of the projection (the `XYZ` part of the `CTYPE`i `= 'ABCD-XYZ'`). The reference point of the projection, $(x, y) = (0, 0)$, is either the north pole of the native spherical coordinates (zenithal projections), the zero point, $(\phi, \theta) = (0, 0)$ (cylindrical and global projections), or the mid-point in conic projections. The zenithal polynomial (`ZPN`) has special requirements.

The mathematical details of these projections and their history and reasons for their use are too extensive to describe here; see the above references. Fig. 2 illustrates the geometry of zenithal projections. The projection plane is tangential to the sphere at the reference point and the projection source is at some distance μ along a vertical from that point. The reference point is the north pole of the native spherical system. Three common geometries are illustrated, with the native spherical system `TAN`, used as a good approximation to optical telescopes, shown on the right. Cylindrical projections are made by wrapping the image plane into a cylinder surrounding the celestial sphere and then projecting the coordinates of the sphere in various ways, not all of which are geometric (Fig. 3). The "linear" projection `CAR` is illustrated and appears to be a serious source of confusion. Images that are "linear" over a fairly large area are frequently, in fact, in a modified cylindrical projection called Sanson-Flamsteed (`SFL`). In that projection, $x = \phi \cos\theta = \phi \cos y$, rather than simply $x = \phi$ of `CAR`.

In conic projections, the image plane is wrapped around the sphere in a conical fashion after which the coordinates are projected and the cone unwrapped (Fig. 4). The native coordinate system is chosen so that

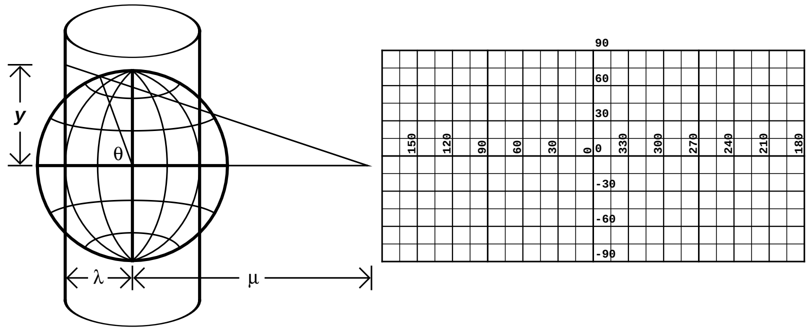


Figure 3: Cylindrical projections: (left) geometry; (right) the plate carrée (CAR) projection native coordinates, no limits.

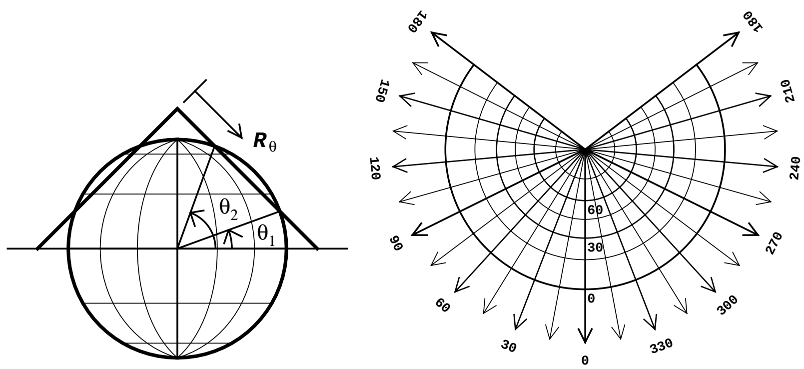


Figure 4: Conic projections: (left) geometry; (right) the conic perspective (COP) projection native coordinates, with $\theta_1 = 20°$, $\theta_2 = 70°$, diverges at $\theta = -45°$.

its pole is parallel to the axis of the cone; the reference point is taken to be at the midpoint $(\phi, \theta) = (0, \theta_{\rm avg})$. The zenithal and cylindrical projections are actually special cases of conic, but treating them separately avoids the mathematical difficulties associated with treating them as conics.

Images of the entire celestial sphere suffer inevitably from a variety of distortions. The `SFL` projection, mentioned above, was developed from cylindrical projections to make the lengths of parallels of latitude be true. The Hammer-Aitoff (`AIT`) projection shown in Fig. 5 is similar but reduces the geometric distortion in polar regions by making the angle between lines of native longitude and latitude more nearly perpendicular. Some of the distortion is then moved to the equator which is no longer evenly divided by the meridians of native longitude. Displaying the whole sphere as six cube faces has been used for analysis of microwave background observations by satellites. The `QSC` projection shown is more mathematically correct (and difficult) than that used by the Cobe satellite. The HEALpix projection (Calabretta & Roukema 2007 [2]) is now the preferred pixelization of cosmic microwave background studies since its organization of the data optimizes spherical harmonic and other analysis methods on the full sphere.

Finally, the native celestial coordinates must be converted to the desired celestial coordinates through a coordinate rotation. The type of celestial coordinate is given by `ABCD` above from a short list of known coordinates including `RA--` and `DEC-` for right ascension and declination, `GLON` and `GLAT` for Galactic longitude and latitude, and so forth. The values of the celestial coordinates at the reference pixel are given by the values of the corresponding `CRVAL`*ia* and are associated with the native coordinates of that point. The coordinate rotation requires one more angle which is taken to be `LONPOLE`*a*, the native longitude of the celestial pole. In a few cases, `LATPOLE`*a*, the native latitude of the celestial pole, must also be given to eliminate multiple

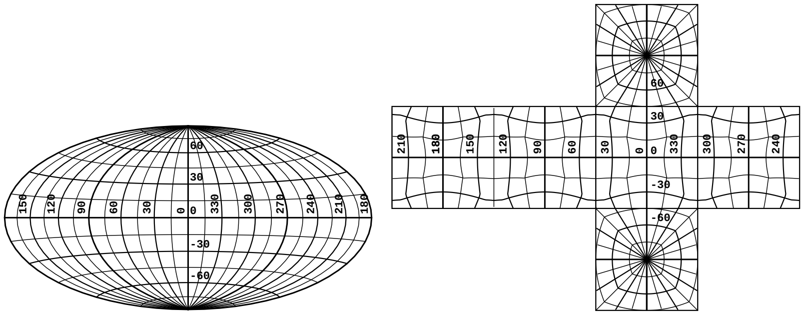


Figure 5: Global projections: (left) Hammer-Aitoff (AIT) projection native coordinates; (right) Quadrilateralized spherical cube (QSC) projection native coordinates.

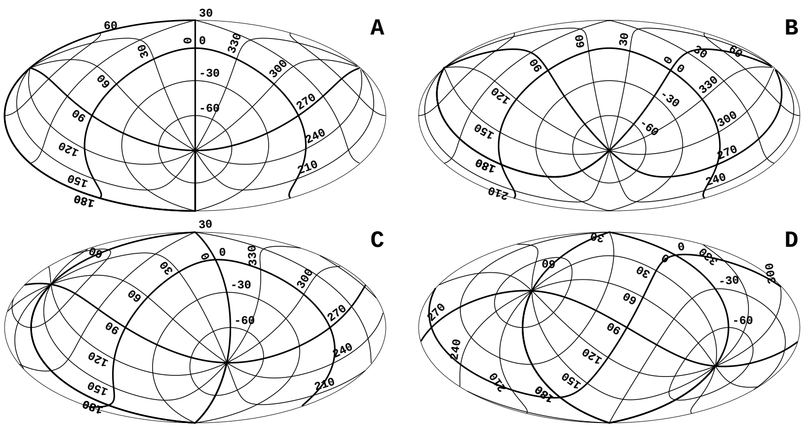


Figure 6: Celestial longitude and latitude plots for the Hammer-Aitoff (AIT) projection with parameters (A) $\alpha_\mathrm{p} = 0$, $\delta_\mathrm{p} = 30°$, $\phi_\mathrm{p} = 180°$; (B) $\alpha_\mathrm{p} = 45°$, $\delta_\mathrm{p} = 30°$, $\phi_\mathrm{p} = 180°$; (C) $\alpha_\mathrm{p} = 0$, $\delta_\mathrm{p} = 30°$, $\phi_\mathrm{p} = 150°$; (D) $\alpha_\mathrm{p} = 0$, $\delta_\mathrm{p} = 30°$, $\phi_\mathrm{p} = 75°$.

solutions of the equations. See Calabretta & Greisen (2002 [9]) for the details of the computation.

Fig. 6 illustrates the effects of different values of the reference coordinates (CRVAL*ia* or (α_0, δ_0)) and LONPOLE (ϕ_p). The first graticule, A, when compared to the non-oblique native coordinate graticule presented earlier (Fig. 5), illustrates the effect of changing δ_p (and hence δ_0). Comparison of graticules A and B shows that changing α_p (and hence α_0) results in a simple change in origin of longitude. Graticules A, C, and D show the more interesting effect of varying ϕ_p. For the zenithal projections the result is indistinguishable from a bulk rotation of the image plane, but this is not the case for any other class of projection.

Additional FITS keywords needed to define celestial coordinates fully include RADESYS*a* to specify the reference system for equatorial or ecliptic coordinates and EQUINOX*a* to specify the epoch of the mean equator and equinox in years. The former keyword has allowed values 'ICRS', 'FK5', and 'GAPPT', plus 'FK4' and FK4-NO-E' which refer to older reference systems whose use is deprecated.

A.4.2 Spectroscopic coordinates

Spectroscopic coordinates have an undeserved reputation for difficulty. Unlike projective celestial coordinates discussed above, there is only one fundamental equation, namely wavelength times frequency equals the speed of light, or $\lambda\nu = c$. All spectroscopic coordinates are simply proportional to λ, ν, or the apparent radial

Table 5: Spectral coordinate type codes, (characters 1–4 of `CTYPE`ia).

Code	Name	Sym	Avar	Units	Formula
`FREQ`	Frequency	ν	`F`	Hz	
`ENER`	Energy	E	`F`	J	$h\nu$
`WAVN`	Wavenumber	κ	`F`	m^{-1}	ν/c
`VRAD`	Radio velocity	V	`F`	$\mathrm{m\ s}^{-1}$	$c(\nu_0 - \nu)/\nu_0$
`WAVE`	Vacuum wavelength	λ	`W`	m	
`VOPT`	Optical velocity	Z	`W`	$\mathrm{m\ s}^{-1}$	$c(\lambda - \lambda_0)/\lambda_0$
`ZOPT`	Redshift	z	`W`	–	$(\lambda - \lambda_0)/\lambda_0$
`AWAV`	Air wavelength	λ_a	`A`	m	
`VELO`	App. radial velocity	v	`V`	$\mathrm{m\ s}^{-1}$	
`BETA`	Beta factor	β	`V`	–	v/c

velocity (v), with the exception of the wavelength in air. These three quantities are related by

$$\nu = \frac{c}{\lambda} \tag{8}$$

$$\nu = \nu_0 \frac{c - v}{\sqrt{c^2 - v^2}} \tag{9}$$

$$\lambda = \frac{c}{\nu} \tag{10}$$

$$\lambda = \lambda_0 \frac{c + v}{\sqrt{c^2 - v^2}} \tag{11}$$

$$v = c \frac{\nu_0^2 - \nu^2}{\nu_0^2 + \nu^2} \tag{12}$$

$$v = c \frac{\lambda^2 - \lambda_0^2}{\lambda^2 + \lambda_0^2} \tag{13}$$

The real difficulties in spectroscopy arise in part from the difficulty of building linear spectrometers at optical and infra-red wavelengths. Non-linear algorithms which approximate real, if somewhat idealized, optical dispersers are described in Greisen *et al.* (2005 [10]) which is the fundamental reference for spectroscopic WCS in FITS. The other significant complication in spectroscopy is the definition of the coordinate frames used to convert observed frequency into the conventional or apparent velocity of the astronomical object. The FITS keywords used to address these issues will be listed below, but discussions of the reference frames and other difficulties related to velocity are to be found in the above reference.

The list of spectroscopic coordinates used in FITS — the first part of the value of `CTYPE`ia `= 'ABCD-XYZ'` — is shown in Table 5. The non-linear algorithm codes used to relate these coordinates have a fixed pattern in which the first character specifies the physical parameter type in which the data are regularly sampled from the list of "associate variables" wavelength (`W`), frequency (`F`), apparent radial velocity (`V`), and wavelength in air (`A`). The second character is `2` and the third character specifies the physical parameter type from the list of associate variables in which the coordinate is expressed. Thus, algorithm code `F2V` is used for data regularly sampled in frequency but with keywords `CUNIT`ia, `CRVAL`ia, and `CDELT`ia or `CD`i_ja expressed in apparent radial velocity (`VELO` or `BETA`) values and units.

Note that velocities require the presence of a reference spectral line to provide the parameter ν_0 or λ_0. These parameters are given in the FITS header with keywords `RESTFRQ`a (in Hz) and `RESTWAV`a (in m), respectively. Keyword `RESTFREQ` has been widely used for the primary WCS (a null). Note that the presence of multiple spectral lines within a single observation is one of the reasons to use alternate WCS descriptions with different values of a.

Velocities are more interesting scientifically if they are corrected at least for the diurnal and annual motions of the observer and perhaps also for the motion of Solar System barycentre and even of the Galaxy. Corrections from one velocity reference frame to another are made from the scalar product of the celestial direction vector and the relative velocity vector of the two reference frames. The FITS keyword `SPECSYS`a gives a character string to specify the reference frame in which velocities are expressed in the HDU header. Since the dot product in the velocity correction makes most velocities a function of celestial coordinate, it is also necessary to specify the velocity reference frame in which velocities are not dependent on coordinate. This keyword, `SSYSOBS`a, has the obvious default value of `'TOPOCENT''` or topocentric. Other allowed values of these keywords include `GEOCENTR'`, `'BARYCENT'`, `'LSRK'` (kinematic local standard of rest), `'LSRD'` (dynamic local standard of rest), `'GALACTOC'`, `'LOCALGRP'`, `'CMBDIPOL'` (cosmic microwave background dipole), and `'SOURCE'`. The last choice allows velocities within an object to be expressed relative to each other, but requires that the source velocity `ZSOURCE`a be given with respect to some other standard of rest specified with `SSYSSRC`a.

In order for software to compute the corrections from topocentric observations to position-dependent velocities with respect to some other standard of rest, it is necessary to provide the date of the mid-point of the observation (`MJD-AVG` or `DATE-AVG`) and the location of the observatory. Rather than depend on one of the very numerous geodetic datums, FITS requests the position in a right-handed, geocentric, Cartesian form using keywords `OBSGEO-X`, `OBSGEO-Y`, and `OBSGEO-Z`. To aid in this computation, the FITS writer is encouraged to provide the relative velocity between the observer and the selected standard of rest in the direction of the celestial reference coordinate using keyword `VELOSYS`a.

A.4.3 Conventional coordinates

The coordinates discussed in the previous two sections have values which are continuous, at least over some defined range. Wells *et al.* (1981 [25]) introduced an additional concept of "conventional" coordinates. These are coordinates whose values can only be integer and hence cannot be interpolated. The reference pixel value `CRVAL`ia and coordinate increment `CDELT`ia must be integers and no other kind of coordinate may be rotated into a conventional coordinate through the `PC`i_ja coordinate rotation matrix. Furthermore, the association of values to these coordinates is solely by convention.

The simplest such coordinate is named `COMPLEX` and has values 1, 2, and 3 for the real part, imaginary part, and weight of a complex number. This axis is widely used for radio interferometric fringe visibility data which are fundamentally complex numbers with an associated weight (σ^{-2}). Another conventional axis, also widely used for radio interferometric data, is the `STOKES` or polarization axis. Conventional values 1 through 4 are used for the standard Stokes I, Q, U, and V polarizations, respectively. Values -1 through -4 are used for the polarization cross products RR, LL, RL, and LR, respectively, from an interferometer observing in circular polarization. For a linearly-polarized instrument, polarization products XX, YY, XY, and YX are assigned values -5 through -8, respectively.

A.4.4 `LOG` and `TAB` projections

Two very general non-linear "projections" were first specified by Greisen *et al.* (2005 [10]). Since they appeared in the spectroscopic WCS paper and have well-known applications to spectroscopy, they have been widely assumed to be relevant only to spectroscopic axes. This is not true; both may be used for any type of coordinate whatsoever.

The first of these is the `LOG` or logarithm, non-linear projection. If x_i is given by Eq. 7, then the physical coordinate S_i on logarithmic axis i is given by

$$S_i = S_{ri} e^{x_i / S_{ri}} , \tag{14}$$

where S_{ri} is the reference pixel value (`CRVAL`ia) on that axis. This form of the logarithm has two desirable attributes. First, the coordinate is roughly linear ($S_i \approx S_{ri} + x_i$) near the reference pixel and, second, the coordinate specifications for `CRVAL`ia, `CDELT`ia, and `CD`i_ja are all in sensible physical units such as `Hz` or `m` as specified in `CUNIT`ia.

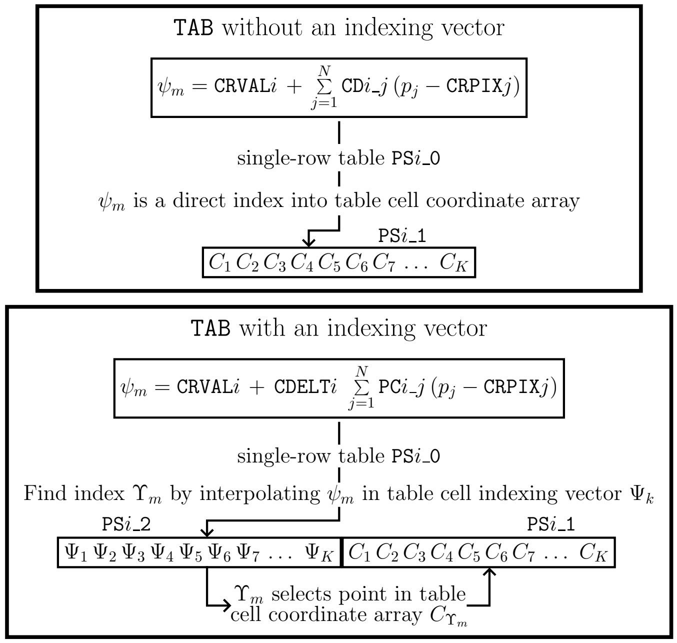


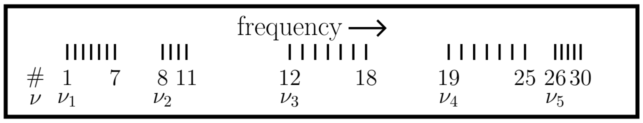
Figure 7: Logic flow in the TAB algorithm with and without an index vector. The coordinate array subscript m associated with intermediate world coordinate axis i is specified with keyword PVi_3. In the case of an independent TAB axis, it would have value 1. Note that ψ_m is x_i computed with Eq. 7.

The second of these non-linear algorithms is a "when all else fails" table look-up to obtain the coordinate values. The definition of this algorithm has an implementation — and perhaps even philosophical — difficulty. The coordinates of the data in one HDU depend, in this algorithm, on the contents of other HDUs, one for each TAB type axis. Despite this difficulty, there is almost no other way to describe, for example, an arbitrary list of the times at which each plane of the image was recorded. One could avoid a TAB axis in this case by writing each separate image in a different FITS file or FITS IMAGE extension. However, this would remove the display and analysis advantages of having all the data available in a single "cube." Many modern instruments are capable of nearly arbitrary placement of their simultaneous recordings along one or more axes such as celestial coordinate and/or frequency or wavelength. Combining the simultaneously recorded data in n-dimensional arrays having table look-up axes may well achieve added value and simplicity during calibration, editing, display, and analysis stages of the subsequent data processing.

The basic concept of the TAB algorithm is rather simple. The intermediate world coordinate x_i is found with Eq. 7. A vector of index values is then searched to locate the two index values surrounding x_i, where the index vector values must be monotonically increasing or decreasing. The linearly interpolated index value is then used as a pointer into an array of coordinate values, where, for example, an index pointer of 7.3 would require a suitable interpolation between the seventh and eighth value in the coordinate array. This basic concept is illustrated symbolically in Fig. 7. All of the complexities in the description of the TAB algorithm in Greisen *et al.* (2005 [10]) come from the extension of this straightforward concept to M related and inseparable axes. Even the subscript m in Fig. 7 arises because axis i is associated with the m^{th} axis of the coordinate array. Here we will ignore this and other practical difficulties; they are addressed fully in Greisen *et al.* (2005 [10]).

The index vector(s) and coordinate array are columns in a BINTABLE HDU in the same FITS file as the HDU that references them. This table has only one data row. The TAB algorithm requires a number of character-valued parameters. PSi_0a is used to specify the EXTNAME of this coordinate HDU, while PSi_1a specifies the column name (TTYPEn_1) of the coordinate array and PSi_2a specifies the column name (TTYPEn_2) of the index array. If the index array column is omitted, the index values are taken to be 1 through K, the number of values in the coordinate array. Note that the dimensions of the index vector(s) and coordinate arrays are

multiple spectral channels
at each of multiple base frequencies



`TAB` parameters and values

`CRVAL`i = `CRPIX`i = `CDELT`i = `PC`i_i = 1

single-row table `PS`i_`0`

values in the cell of the indexing vector `PS`i_`2` =

$\boxed{1, 7, 8, 11, 12, 18, 19, 25, 26, 30}$

corresponding values
in the cell of the coordinate array `PS`i_`1` =

$\boxed{\nu_1,\ \nu_1+6\delta_1,\ \nu_2,\ \nu_2+3\delta_2,\ \nu_3,\ \nu_3+6\delta_3,\ \nu_4,\ \nu_4+6\delta_4,\ \nu_5,\ \nu_5+4\delta_5}$

Figure 8: Example taken from radio interferometry using `TAB` with an indexing vector. The FITS keywords shown are suitable for the random groups format. The observation is made at a number of frequencies, with a number N_ℓ of regularly spaced (by δ_ℓ) spectral channels beginning from each of a number of arbitrary base frequencies ν_ℓ. The use of an indexing vector reduces the number of values in the table from one array of 30 to two arrays of 10 each. In a real case the number of spectral channels would be significantly larger, making the space savings significant. In this example, pixel $p_j = 6$ produces $\psi_m = 6$. This lies at $\Upsilon_m = 1\frac{5}{6}$. The resulting coordinate is then $\frac{1}{6}\nu_1 + \frac{5}{6}(\nu_1 + 6\delta_1)$ which, as one would expect, equals $\nu_1 + 5\delta_1$. Note that this example involves only a single independent `TAB` axis, so that `PV`i_`3` $= m \equiv 1$.

specified in the header of the coordinate `BINTABLE` and do not need to be specified in the HDU which is to use the coordinate table.

Let us end this section with a simple numerical example. The Expanded Very Large Array has a new correlator which will allow up to 64 groups of spectral channels to be placed at more-or-less independent frequencies and independent channel separations within the very wide bands of the front-end receivers. The number of channels involved are in the thousands, but let us take much smaller numbers for simplicity. Fig. 8 shows 5 frequency bands with frequencies ν_ℓ which may be in any order, but are plotted for simplicity as if they were monotonically increasing. Each band has a number of spectral channels separated by δ_ℓ. The index vector is monotonic, as is required, and increasing. The intermediate world coordinates x_i have values from 1 through N, the total number of spectral channels. An example of the computation is carried out in the figure caption.

B Example FITS file headers

The sample FITS header-data-units (HDUs) listed below come from a variety of data files chosen more or less at random. The only changes made were to omit most HISTORY and commentary card images and drop multiple versions of tables, keeping only the first.

In image format, all of the image data are in binary records recorded immediately after the primary HDU. In the random groups format, all of the visibility data are in binary records recorded after the primary HDU and before any of the tables. In the table form, the small tables follow primary HDU and the the *uv*-data table HDU and data come last.

B.1 Primary HDU of Image File

```
SIMPLE  =                    T /
BITPIX  =                  -32 /
NAXIS   =                    4 /
NAXIS1  =                  512 /
NAXIS2  =                  512 /
NAXIS3  =                    1 /
NAXIS4  =                    1 /
EXTEND  =                    T /Tables following main image
BLOCKED =                    T /Tape may be blocked
OBJECT  = 'NGC6503 '           /Source name
TELESCOP= 'VLA     '           /
INSTRUME= 'VLA     '           /
OBSERVER= 'AM510   '           /
DATE-OBS= '1996-02-27'         /Obs start date YYYY-MM-DD
DATE-MAP= '2011-10-25'         /Last processing date YYYY-MM-DD
BSCALE  =    1.00000000000E+00 /REAL = TAPE * BSCALE + BZERO
BZERO   =    0.00000000000E+00 /
BUNIT   = 'JY/BEAM '           /Units of flux
EQUINOX =      2.000000000E+03 /Epoch of RA DEC
VELREF  =                    2 />256 RADIO, 1 LSR 2 HEL 3 OBS
ALTRVAL =    2.60000000000E+04 /Altenate FREQ/VEL ref value
ALTRPIX =      9.609375000E-01 /Altenate FREQ/VEL ref pixel
OBSRA   =    2.67250000000E+02 /Antenna pointing RA
OBSDEC  =    7.01166666667E+01 /Antenna pointing DEC
RESTFREQ=    1.42040575200E+09 /Rest frequency
DATAMAX =      3.677410185E-01 /Maximum pixel value
DATAMIN =     -1.565772691E-03 /Minimum pixel value
CTYPE1  = 'RA---SIN'           /
CRVAL1  =    2.67250000000E+02 /
CDELT1  =     -8.333333535E-04 /
CRPIX1  =      2.560000000E+02 /
CROTA1  =      0.000000000E+00 /
CTYPE2  = 'DEC--SIN'           /
CRVAL2  =    7.01166666667E+01 /
CDELT2  =      8.333333535E-04 /
CRPIX2  =      2.570000000E+02 /
CROTA2  =      0.000000000E+00 /
CTYPE3  = 'FREQ    '           /
CRVAL3  =    1.42033693682E+09 /
CDELT3  =      2.441406250E+04 /
CRPIX3  =      1.000000000E+00 /
CROTA3  =      0.000000000E+00 /
CTYPE4  = 'STOKES  '           /
CRVAL4  =    1.00000000000E+00 /
```

```
CDELT4  =      1.000000000E+00 /
CRPIX4  =      1.000000000E+00 /
CROTA4  =      0.000000000E+00 /
HISTORY AIPS HEADER2  WTNOISE  =  1.192631721E+00 /AIPS Catalog Header Keyword
HISTORY AIPS HEADER2  SUMWTIN  =  2.456301500E+06 /AIPS Catalog Header Keyword
HISTORY AIPS HEADER2  CCFLUX   =  5.948930383E-01 /AIPS Catalog Header Keyword
HISTORY AIPS HEADER2  CCTOTAL  =  5.948930383E-01 /AIPS Catalog Header Keyword
ORIGIN  = 'AIPSprimate      COAOARN             31DEC12'    /
DATE    = '2012-01-18'   / File written on Greenwich yyyy-mm-dd
HISTORY AIPS   IMNAME='NGC6503 full' IMCLASS='ICL001' IMSEQ=   1     /
HISTORY AIPS   USERNO=   36            /
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
HISTORY AIPS   CLEAN BMAJ=  3.8286E-03 BMIN=  3.6046E-03 BPA= -82.27
HISTORY AIPS   CLEAN NITER=     1267 PRODUCT=1   / NORMAL
HISTORY AIPS   IMAGE ITYPE=2 XPOFF=  0.00000000E+00 YPOFF=  0.00000000E+00
END
```

B.2 Primary HDU of Random Groups

```
SIMPLE  =                    T /
BITPIX  =                  -32 /
NAXIS   =                    7 /
NAXIS1  =                    0 /No standard image just group
NAXIS2  =                    3 /
NAXIS3  =                    4 /
NAXIS4  =                   64 /
NAXIS5  =                    2 /
NAXIS6  =                    1 /
NAXIS7  =                    1 /
EXTEND  =                    T /This is the antenna file
BLOCKED =                    T /Tape may be blocked
OBJECT  = 'MULTI   '           /Source name
TELESCOP= 'EVLA    '           /
INSTRUME= 'EVLA    '           /
OBSERVER= 'TOSR0001'           /
DATE-OBS= '2010-03-08'         /Obs start date YYYY-MM-DD
DATE-MAP= '2010-03-08'         /Last processing date YYYY-MM-DD
BSCALE  =    1.00000000000E+00 /REAL = TAPE * BSCALE + BZERO
BZERO   =    0.00000000000E+00 /
BUNIT   = 'UNCALIB '           /Units of flux
EQUINOX =      2.000000000E+03 /Epoch of RA DEC
ALTRPIX =      3.300000000E+01 /Altenate FREQ/VEL ref pixel
CTYPE2  = 'COMPLEX '           /1=real,2=imag,3=weight
CRVAL2  =    1.00000000000E+00 /
CDELT2  =      1.000000000E+00 /
CRPIX2  =      1.000000000E+00 /
CROTA2  =      0.000000000E+00 /
CTYPE3  = 'STOKES  '           /-1=RR, -2=LL, -3=RL, -4=LR
CRVAL3  =   -1.00000000000E+00 /
CDELT3  =     -1.000000000E+00 /
CRPIX3  =      1.000000000E+00 /
CROTA3  =      0.000000000E+00 /
CTYPE4  = 'FREQ    '           /Frequency in Hz.
CRVAL4  =    4.89600000000E+09 /
CDELT4  =      2.000000000E+06 /
CRPIX4  =      3.300000000E+01 /
CROTA4  =      0.000000000E+00 /
```

```
CTYPE5  = 'IF      '           /Freq. group no. in CH table
CRVAL5  =    1.00000000000E+00 /
CDELT5  =      1.000000000E+00 /
CRPIX5  =      1.000000000E+00 /
CROTA5  =      0.000000000E+00 /
CTYPE6  = 'RA      '           /Right Ascension in deg.
CRVAL6  =    0.00000000000E+00 /
CDELT6  =      1.000000000E+00 /
CRPIX6  =      1.000000000E+00 /
CROTA6  =      0.000000000E+00 /
CTYPE7  = 'DEC     '           /Declination in deg.
CRVAL7  =    0.00000000000E+00 /
CDELT7  =      1.000000000E+00 /
CRPIX7  =      1.000000000E+00 /
CROTA7  =      0.000000000E+00 /
GROUPS  =                    T /
GCOUNT  =               609900 /
PCOUNT  =                    9 /
PTYPE1  = 'UU--    '
PSCAL1  =    2.04248366013E-10 /
PZERO1  =    0.00000000000E+00 /
PTYPE2  = 'VV--    '
PSCAL2  =    2.04248366013E-10 /
PZERO2  =    0.00000000000E+00 /
PTYPE3  = 'WW--    '
PSCAL3  =    2.04248366013E-10 /
PZERO3  =    0.00000000000E+00 /
PTYPE4  = 'DATE    '
PSCAL4  =    1.00000000000E+00 /
PZERO4  =    2.45526350000E+06 /
PTYPE5  = 'DATE    '
PSCAL5  =    1.00000000000E+00 /
PZERO5  =    0.00000000000E+00 /
PTYPE6  = 'BASELINE'
PSCAL6  =    1.00000000000E+00 /
PZERO6  =    0.00000000000E+00 /
PTYPE7  = 'FREQSEL '
PSCAL7  =    1.00000000000E+00 /
PZERO7  =    0.00000000000E+00 /
PTYPE8  = 'SOURCE  '
PSCAL8  =    1.00000000000E+00 /
PZERO8  =    0.00000000000E+00 /
PTYPE9  = 'INTTIM  '
PSCAL9  =    1.00000000000E+00 /
PZERO9  =    0.00000000000E+00 /
         / Where baseline = 256*ant1 + ant2 + (array#-1)/100
HISTORY FITTP DATAOUT = 'MYFITS:FITTP.117' / data written to disk file
ORIGIN  = 'AIPSprimate      COAOARN              31DEC12'    /
DATE    = '2012-01-05'   / File written on Greenwich yyyy-mm-dd
HISTORY AIPS   IMNAME='Empty field ' IMCLASS='UVDATA' IMSEQ=   1     /
HISTORY AIPS   USERNO= 2010            /
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
HISTORY AIPS   SORT ORDER = 'TB'
              / Where T means TIME (IAT)
              / Where B means BASELINE NUM
HISTORY AIPS WTSCAL =  1.00000000000E+00 / CMPLX WTS=WTSCAL*(TAPE*BSCALE+BZERO)
END
```

B.3 Primary HDU of *uv* table form

```
SIMPLE  =                    T / Standard FITS file
BITPIX  =                    8 /
NAXIS   =                    2 /
NAXIS1  =            777777701 / Signature code for UV data in table
NAXIS2  =                    0 / No data in primary array
EXTEND  =                    T / All data in tables
BLOCKED =                    T / Tape may be blocked
HISTORY FITAB DATAOUT = 'MYFITS:FITAB.117' / data written to disk file
ORIGIN  = 'AIPSprimate      COAOARN              31DEC12'    /
DATE    = '2012-01-05'   / File written on Greenwich yyyy-mm-dd
HISTORY AIPS   IMNAME='Empty field ' IMCLASS='UVDATA' IMSEQ=   1     /
HISTORY AIPS   USERNO= 2010            /
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
         / Where baseline = 256*ant1 + ant2 + (array#-1)/100
HISTORY AIPS   SORT ORDER = 'TB'
              / Where T means TIME (IAT)
              / Where B means BASELINE NUM
HISTORY AIPS   IPIECE= 1 NPIECE= 1     / piece number
HISTORY AIPS   FIRSTVIS=           1     / first vis #
HISTORY AIPS   LASTVIS =      609900     / last vis #
END
```

B.4 Table HDUs initially input to AIPS

B.4.1 AIPS AN tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   66 / Width of table in bytes
NAXIS2  =                   25 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   14 / Number of fields in each row
EXTNAME = 'AIPS AN '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '8A      '           / FORTRAN format of field  1
TTYPE1  = 'ANNAME          '   / Type (heading) of field  1
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '3D      '           / FORTRAN format of field  2
TTYPE2  = 'STABXYZ         '   / Type (heading) of field  2
TUNIT2  = 'METERS  '           / Physical units of field  2
TFORM3  = '0D      '           / FORTRAN format of field  3
TTYPE3  = 'ORBPARM         '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'NOSTA           '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'MNTSTA          '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1E      '           / FORTRAN format of field  6
TTYPE6  = 'STAXOF          '   / Type (heading) of field  6
TUNIT6  = 'METERS  '           / Physical units of field  6
TFORM7  = '1E      '           / FORTRAN format of field  7
```

```
TTYPE7  = 'DIAMETER        '   / Type (heading) of field  7
TUNIT7  = 'METERS  '           / Physical units of field  7
TFORM8  = '2E      '           / FORTRAN format of field  8
TTYPE8  = 'BEAMFWHM        '   / Type (heading) of field  8
TUNIT8  = 'DEGR/M  '           / Physical units of field  8
TFORM9  = '1A      '           / FORTRAN format of field  9
TTYPE9  = 'POLTYA          '   / Type (heading) of field  9
TUNIT9  = '        '           / Physical units of field  9
TFORM10 = '1E      '           / FORTRAN format of field 10
TTYPE10 = 'POLAA           '   / Type (heading) of field 10
TUNIT10 = 'DEGREES '           / physical units of field 10
TFORM11 = '0E      '           / FORTRAN format of field 11
TTYPE11 = 'POLCALA         '   / Type (heading) of field 11
TUNIT11 = '        '           / physical units of field 11
TFORM12 = '1A      '           / FORTRAN format of field 12
TTYPE12 = 'POLTYB          '   / Type (heading) of field 12
TUNIT12 = '        '           / physical units of field 12
TFORM13 = '1E      '           / FORTRAN format of field 13
TTYPE13 = 'POLAB           '   / Type (heading) of field 13
TUNIT13 = 'DEGREES '           / physical units of field 13
TFORM14 = '0E      '           / FORTRAN format of field 14
TTYPE14 = 'POLCALB         '   / Type (heading) of field 14
TUNIT14 = '        '           / physical units of field 14
ARRAYX  =   0.00000000000000000E+00
ARRAYY  =   0.00000000000000000E+00
ARRAYZ  =   0.00000000000000000E+00
GSTIA0  =   0.16559053797569999E+03
DEGPDY  =   0.36098564385930001E+03
FREQ    =   0.48960000000000000E+10
RDATE   = '2010-03-08'
POLARX  =  -0.44948905706405640E-01
POLARY  =   0.26670348644256592E+00
UT1UTC  =   0.43832778930664062E-01
DATUTC  =   0.00000000000000000E+00
TIMSYS  = 'UTC     '
ARRNAM  = 'EVLA    '
XYZHAND = 'RIGHT   '
FRAME   = 'ITRF    '
NUMORB  =            0
NOPCAL  =            0
NO_IF   =            2
FREQID  =         -999
IATUTC  =   0.34000000000000000E+02
POLTYPE = '        '
END
```

B.4.2 AIPS CD tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  268 / Width of table in bytes
NAXIS2  =                   28 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    7 / Number of fields in each row
EXTNAME = 'AIPS CD '           / AIPS table file
```

```
EXTVER  =                    1 / Version number of table
TFORM1  = '1J      '           / FORTRAN format of field  1
TTYPE1  = 'ANTENNA NO.     '   / Type (heading) of field  1
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '1J      '           / FORTRAN format of field  2
TTYPE2  = 'SUBARRAY        '   / Type (heading) of field  2
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'FREQ ID         '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '16E     '           / FORTRAN format of field  4
TTYPE4  = 'TCAL1           '   / Type (heading) of field  4
TUNIT4  = 'KELVIN  '           / Physical units of field  4
TFORM5  = '16E     '           / FORTRAN format of field  5
TTYPE5  = 'SOLCAL1         '   / Type (heading) of field  5
TUNIT5  = 'KELVIN  '           / Physical units of field  5
TFORM6  = '16E     '           / FORTRAN format of field  6
TTYPE6  = 'TCAL2           '   / Type (heading) of field  6
TUNIT6  = 'KELVIN  '           / Physical units of field  6
TFORM7  = '16E     '           / FORTRAN format of field  7
TTYPE7  = 'SOLCAL2         '   / Type (heading) of field  7
TUNIT7  = 'KELVIN  '           / Physical units of field  7
NO_IF   =           16
NO_POL  =            2
NO_ANT  =           28
REVISION=            1
RDATE   = '2017-05-11'
END
```

B.4.3 AIPS CQ tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   68 / Width of table in bytes
NAXIS2  =                    1 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   14 / Number of fields in each row
EXTNAME = 'AIPS CQ '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1J      '           / FORTRAN format of field  1
TTYPE1  = 'FRQSEL          '   / Type (heading) of field  1
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '1J      '           / FORTRAN format of field  2
TTYPE2  = 'SUBARRAY        '   / Type (heading) of field  2
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'FFT_SIZE        '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'NO_CHAN         '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'SPEC_AVG        '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1D      '           / FORTRAN format of field  6
TTYPE6  = 'EDGE_FRQ        '   / Type (heading) of field  6
```

```
TUNIT6  = 'HZ      '            / Physical units of field  6
TFORM7  = '1D      '            / FORTRAN format of field  7
TTYPE7  = 'CHAN_BW         '    / Type (heading) of field  7
TUNIT7  = 'HZ      '            / Physical units of field  7
TFORM8  = '8A      '            / FORTRAN format of field  8
TTYPE8  = 'TAPER_FN        '    / Type (heading) of field  8
TUNIT8  = '        '            / Physical units of field  8
TFORM9  = '1J      '            / FORTRAN format of field  9
TTYPE9  = 'OVR_SAMP        '    / Type (heading) of field  9
TUNIT9  = '        '            / Physical units of field  9
TFORM10 = '1J      '            / FORTRAN format of field 10
TTYPE10 = 'ZERO_PAD        '    / Type (heading) of field 10
TUNIT10 = '        '            / physical units of field 10
TFORM11 = '1J      '            / FORTRAN format of field 11
TTYPE11 = 'FILTER          '    / Type (heading) of field 11
TUNIT11 = '        '            / physical units of field 11
TFORM12 = '1E      '            / FORTRAN format of field 12
TTYPE12 = 'TIME_AVG        '    / Type (heading) of field 12
TUNIT12 = 'SECONDS '            / physical units of field 12
TFORM13 = '1J      '            / FORTRAN format of field 13
TTYPE13 = 'NO_BITS         '    / Type (heading) of field 13
TUNIT13 = '        '            / physical units of field 13
TFORM14 = '1J      '            / FORTRAN format of field 14
TTYPE14 = 'FFT_OVLP        '    / Type (heading) of field 14
TUNIT14 = '        '            / physical units of field 14
NO_IF   =            1
TABREV  =            1
END
```

B.4.4 AIPS CT tables

```
XTENSION= 'BINTABLE'            / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   98 / Width of table in bytes
NAXIS2  =                   55 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   12 / Number of fields in each row
EXTNAME = 'AIPS CT '            / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '            / FORTRAN format of field  1
TTYPE1  = 'TIME            '    / Type (heading) of field  1
TUNIT1  = 'DAYS    '            / Physical units of field  1
TFORM2  = '1D      '            / FORTRAN format of field  2
TTYPE2  = 'UT1-UTC         '    / Type (heading) of field  2
TUNIT2  = 'SECONDS '            / Physical units of field  2
TFORM3  = '1D      '            / FORTRAN format of field  3
TTYPE3  = 'IAT-UTC         '    / Type (heading) of field  3
TUNIT3  = 'SECONDS '            / Physical units of field  3
TFORM4  = '1D      '            / FORTRAN format of field  4
TTYPE4  = 'A1-IAT          '    / Type (heading) of field  4
TUNIT4  = 'SECONDS '            / Physical units of field  4
TFORM5  = '1A      '            / FORTRAN format of field  5
TTYPE5  = 'UT1 TYPE        '    / Type (heading) of field  5
TUNIT5  = '        '            / Physical units of field  5
TFORM6  = '2D      '            / FORTRAN format of field  6
TTYPE6  = 'WOBXY           '    / Type (heading) of field  6
```

```
TUNIT6  = 'ARCSEC  '           / Physical units of field  6
TFORM7  = '1A      '           / FORTRAN format of field  7
TTYPE7  = 'WOB TYPE        '   / Type (heading) of field  7
TUNIT7  = '        '           / Physical units of field  7
TFORM8  = '1D      '           / FORTRAN format of field  8
TTYPE8  = 'DPSI            '   / Type (heading) of field  8
TUNIT8  = 'RADIANS '           / Physical units of field  8
TFORM9  = '1D      '           / FORTRAN format of field  9
TTYPE9  = 'DDPSI           '   / Type (heading) of field  9
TUNIT9  = 'RAD/SEC '           / Physical units of field  9
TFORM10 = '1D      '           / FORTRAN format of field 10
TTYPE10 = 'DEPS            '   / Type (heading) of field 10
TUNIT10 = 'RADIANS '           / physical units of field 10
TFORM11 = '1D      '           / FORTRAN format of field 11
TTYPE11 = 'DDEPS           '   / Type (heading) of field 11
TUNIT11 = 'RAD/SEC '           / physical units of field 11
TFORM12 = '2D      '           / FORTRAN format of field 12
TTYPE12 = 'TIME INT        '   / Type (heading) of field 12
TUNIT12 = 'DAYS    '           / physical units of field 12
OBSCODE = 'BR129   '
RDATE   = '2008-04-24'
NO_STKD =            2
STK_1   =           -1
NO_BAND =            1
NO_CHAN =          256
REF_FREQ=   0.12174250000000000E+11
CHAN_BW =   0.15625000000000000E+05
REF_PIXL=   0.10000000000000000E+01
TABREV  =            3
C_SRVR  = 'kepler  '
C_VERSN = '9.1     '
A_VERSN = '2.2     '
I_VERSN = '0.0     '
E_VERSN = '9.1     '
ACCELGRV=   0.97803184600000002E+01
E-FLAT  =   0.33528099999999999E-02
EARTHRAD=   0.63781370000000000E+07
MMSEMS  =   0.12300020000000000E-01
EPHEPOC =         2000
ETIDELAG=   0.00000000000000000E+00
GAUSS   =   0.17202098950000001E-01
GMMOON  =   0.49027975000000000E+13
GMSUN   =   0.13271243799999993E+21
LOVE_H  =   0.60967000000000005E+00
LOVE_L  =   0.85199999999999984E-01
PRE_DATA=   0.50290965999999999E+04
REL_DATA=   0.10000000000000000E+01
TIDALUT1=            0
TSECAU  =   0.49900478199999998E+03
U-GRV-CN=   0.66720000000000016E-10
VLIGHT  =   0.29979245800000000E+09
END
```

B.4.5 AIPS FG tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   69 / Width of table in bytes
NAXIS2  =                 5575 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    9 / Number of fields in each row
EXTNAME = 'AIPS FG '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1J      '           / FORTRAN format of field  1
TTYPE1  = 'SOURCE          '   / Type (heading) of field  1
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '1J      '           / FORTRAN format of field  2
TTYPE2  = 'SUBARRAY        '   / Type (heading) of field  2
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'FREQ ID         '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '2J      '           / FORTRAN format of field  4
TTYPE4  = 'ANTS            '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '2E      '           / FORTRAN format of field  5
TTYPE5  = 'TIME RANGE      '   / Type (heading) of field  5
TUNIT5  = 'DAYS    '           / Physical units of field  5
TFORM6  = '2J      '           / FORTRAN format of field  6
TTYPE6  = 'IFS             '   / Type (heading) of field  6
TUNIT6  = '        '           / Physical units of field  6
TFORM7  = '2J      '           / FORTRAN format of field  7
TTYPE7  = 'CHANS           '   / Type (heading) of field  7
TUNIT7  = '        '           / Physical units of field  7
TFORM8  = '4X      '           / FORTRAN format of field  8
TTYPE8  = 'PFLAGS          '   / Type (heading) of field  8
TUNIT8  = '        '           / Physical units of field  8
TFORM9  = '24A     '           / FORTRAN format of field  9
TTYPE9  = 'REASON          '   / Type (heading) of field  9
TUNIT9  = '        '           / Physical units of field  9
ISORTORD=                   5
END
```

B.4.6 AIPS FO tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   36 / Width of table in bytes
NAXIS2  =                 1450 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    7 / Number of fields in each row
EXTNAME = 'AIPS FO '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
```

```
TTYPE2  = 'TIME INTERVAL   '   / Type (heading) of field  2
TUNIT2  = 'DAYS    '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'SOURCE ID       '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'ANTENNA NO.     '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'SUBARRAY        '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1J      '           / FORTRAN format of field  6
TTYPE6  = 'FREQ ID         '   / Type (heading) of field  6
TUNIT6  = '        '           / Physical units of field  6
TFORM7  = '2E      '           / FORTRAN format of field  7
TTYPE7  = 'DOPPOFF         '   / Type (heading) of field  7
TUNIT7  = 'HZ      '           / Physical units of field  7
NO_ANT  =           29
NO_IF   =            2
REVISION=            0
END
```

B.4.7 AIPS FQ tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   44 / Width of table in bytes
NAXIS2  =                    1 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    5 / Number of fields in each row
EXTNAME = 'AIPS FQ '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1J      '           / FORTRAN format of field  1
TTYPE1  = 'FRQSEL          '   / Type (heading) of field  1
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '2D      '           / FORTRAN format of field  2
TTYPE2  = 'IF FREQ         '   / Type (heading) of field  2
TUNIT2  = 'HZ      '           / Physical units of field  2
TFORM3  = '2E      '           / FORTRAN format of field  3
TTYPE3  = 'CH WIDTH        '   / Type (heading) of field  3
TUNIT3  = 'HZ      '           / Physical units of field  3
TFORM4  = '2E      '           / FORTRAN format of field  4
TTYPE4  = 'TOTAL BANDWIDTH '   / Type (heading) of field  4
TUNIT4  = 'HZ      '           / Physical units of field  4
TFORM5  = '2J      '           / FORTRAN format of field  5
TTYPE5  = 'SIDEBAND        '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
NO_IF   =            2
END
```

B.4.8 AIPS IM tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  436 / Width of table in bytes
NAXIS2  =                 1337 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   20 / Number of fields in each row
EXTNAME = 'AIPS IM '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'TIME_INTERVAL   '   / Type (heading) of field  2
TUNIT2  = 'DAYS    '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'SOURCE_ID       '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'ANTENNA_NO      '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'ARRAY           '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1J      '           / FORTRAN format of field  6
TTYPE6  = 'FREQID          '   / Type (heading) of field  6
TUNIT6  = '        '           / Physical units of field  6
TFORM7  = '1E      '           / FORTRAN format of field  7
TTYPE7  = 'I.FAR.ROT       '   / Type (heading) of field  7
TUNIT7  = 'RAD/M**2'           / Physical units of field  7
TFORM8  = '1E      '           / FORTRAN format of field  8
TTYPE8  = 'FREQ.VAR        '   / Type (heading) of field  8
TUNIT8  = 'HZ      '           / Physical units of field  8
TFORM9  = '6D      '           / FORTRAN format of field  9
TTYPE9  = 'PDELAY_1        '   / Type (heading) of field  9
TUNIT9  = 'SECONDS '           / Physical units of field  9
TFORM10 = '6D      '           / FORTRAN format of field 10
TTYPE10 = 'GDELAY_1        '   / Type (heading) of field 10
TUNIT10 = 'SECONDS '           / physical units of field 10
TFORM11 = '6D      '           / FORTRAN format of field 11
TTYPE11 = 'PRATE_1         '   / Type (heading) of field 11
TUNIT11 = 'HZ      '           / physical units of field 11
TFORM12 = '6D      '           / FORTRAN format of field 12
TTYPE12 = 'GRATE_1         '   / Type (heading) of field 12
TUNIT12 = 'SEC/SEC '           / physical units of field 12
TFORM13 = '1E      '           / FORTRAN format of field 13
TTYPE13 = 'DISP_1          '   / Type (heading) of field 13
TUNIT13 = 'SECONDS '           / physical units of field 13
TFORM14 = '1E      '           / FORTRAN format of field 14
TTYPE14 = 'DDISP_1         '   / Type (heading) of field 14
TUNIT14 = 'SEC/SEC '           / physical units of field 14
TFORM15 = '6D      '           / FORTRAN format of field 15
TTYPE15 = 'PDELAY_2        '   / Type (heading) of field 15
TUNIT15 = 'SECONDS '           / physical units of field 15
TFORM16 = '6D      '           / FORTRAN format of field 16
TTYPE16 = 'GDELAY_2        '   / Type (heading) of field 16
```

```
TUNIT16 = 'SECONDS '           / physical units of field 16
TFORM17 = '6D      '           / FORTRAN format of field 17
TTYPE17 = 'PRATE_2         '   / Type (heading) of field 17
TUNIT17 = 'HZ      '           / physical units of field 17
TFORM18 = '6D      '           / FORTRAN format of field 18
TTYPE18 = 'GRATE_2         '   / Type (heading) of field 18
TUNIT18 = 'SEC/SEC '           / physical units of field 18
TFORM19 = '1E      '           / FORTRAN format of field 19
TTYPE19 = 'DISP_2          '   / Type (heading) of field 19
TUNIT19 = 'SECONDS '           / physical units of field 19
TFORM20 = '1E      '           / FORTRAN format of field 20
TTYPE20 = 'DDISP_2         '   / Type (heading) of field 20
TUNIT20 = 'SEC/SEC '           / physical units of field 20
RDATE   = '2008-04-24'
OBSCODE = 'BR129   '
NO_STKD =            2
STK_1   =           -1
NO_BAND =            1
NO_CHAN =          256
REF_FREQ=   0.12174250000000000E+11
CHAN_BW =   0.15625000000000000E+05
REF_PIXL=   0.10000000000000000E+01
NO_POL  =            2
NPOLY   =            6
REVISION=   0.10000000000000000E+01
TABREV  =            2
END
```

B.4.9 AIPS MC tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  120 / Width of table in bytes
NAXIS2  =                 3534 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   21 / Number of fields in each row
EXTNAME = 'AIPS MC '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1J      '           / FORTRAN format of field  2
TTYPE2  = 'SOURCE_ID       '   / Type (heading) of field  2
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'ANTENNA_NO      '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'ARRAY           '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'FREQID          '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1D      '           / FORTRAN format of field  6
TTYPE6  = 'ATMOS           '   / Type (heading) of field  6
TUNIT6  = 'SECONDS '           / Physical units of field  6
```

```
TFORM7  = '1D      '           / FORTRAN format of field  7
TTYPE7  = 'DATMOS          '   / Type (heading) of field  7
TUNIT7  = 'SEC/SEC '           / Physical units of field  7
TFORM8  = '1D      '           / FORTRAN format of field  8
TTYPE8  = 'GDELAY          '   / Type (heading) of field  8
TUNIT8  = 'SECONDS '           / Physical units of field  8
TFORM9  = '1D      '           / FORTRAN format of field  9
TTYPE9  = 'GRATE           '   / Type (heading) of field  9
TUNIT9  = 'SEC/SEC '           / Physical units of field  9
TFORM10 = '1D      '           / FORTRAN format of field 10
TTYPE10 = 'CLOCK_1         '   / Type (heading) of field 10
TUNIT10 = 'SECONDS '           / physical units of field 10
TFORM11 = '1D      '           / FORTRAN format of field 11
TTYPE11 = 'DCLOCK_1        '   / Type (heading) of field 11
TUNIT11 = 'SEC/SEC '           / physical units of field 11
TFORM12 = '1E      '           / FORTRAN format of field 12
TTYPE12 = 'LO_OFFSET_1     '   / Type (heading) of field 12
TUNIT12 = 'HZ      '           / physical units of field 12
TFORM13 = '1E      '           / FORTRAN format of field 13
TTYPE13 = 'DLO_OFFSET_1    '   / Type (heading) of field 13
TUNIT13 = 'HZ/SEC  '           / physical units of field 13
TFORM14 = '1E      '           / FORTRAN format of field 14
TTYPE14 = 'DISP_1          '   / Type (heading) of field 14
TUNIT14 = 'SECONDS '           / physical units of field 14
TFORM15 = '1E      '           / FORTRAN format of field 15
TTYPE15 = 'DDISP_1         '   / Type (heading) of field 15
TUNIT15 = 'SEC/SEC '           / physical units of field 15
TFORM16 = '1D      '           / FORTRAN format of field 16
TTYPE16 = 'CLOCK_2         '   / Type (heading) of field 16
TUNIT16 = 'SECONDS '           / physical units of field 16
TFORM17 = '1D      '           / FORTRAN format of field 17
TTYPE17 = 'DCLOCK_2        '   / Type (heading) of field 17
TUNIT17 = 'SEC/SEC '           / physical units of field 17
TFORM18 = '1E      '           / FORTRAN format of field 18
TTYPE18 = 'LO_OFFSET_2     '   / Type (heading) of field 18
TUNIT18 = 'HZ      '           / physical units of field 18
TFORM19 = '1E      '           / FORTRAN format of field 19
TTYPE19 = 'DLO_OFFSET_2    '   / Type (heading) of field 19
TUNIT19 = 'HZ/SEC  '           / physical units of field 19
TFORM20 = '1E      '           / FORTRAN format of field 20
TTYPE20 = 'DISP_2          '   / Type (heading) of field 20
TUNIT20 = 'SECONDS '           / physical units of field 20
TFORM21 = '1E      '           / FORTRAN format of field 21
TTYPE21 = 'DDISP_2         '   / Type (heading) of field 21
TUNIT21 = 'SEC/SEC '           / physical units of field 21
OBSCODE = 'BR129   '
RDATE   = '2008-04-24'
NO_STKD =            2
STK_1   =           -1
NO_BAND =            1
NO_CHAN =          256
REF_FREQ=   0.12174250000000000E+11
CHAN_BW =   0.15625000000000000E+05
REF_PIXL=   0.10000000000000000E+01
NO_POL  =            2
FFT_SIZE=          512
OVERSAMP=            0
ZERO_PAD=            0
TAPER_FN= 'UNIFORM '
```

```
TABREV  =            1
END
```

B.4.10 AIPS NX tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   28 / Width of table in bytes
NAXIS2  =                   15 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    7 / Number of fields in each row
EXTNAME = 'AIPS NX '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1E      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'TIME INTERVAL   '   / Type (heading) of field  2
TUNIT2  = 'DAYS    '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'SOURCE ID       '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'SUBARRAY        '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'START VIS       '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1J      '           / FORTRAN format of field  6
TTYPE6  = 'END VIS         '   / Type (heading) of field  6
TUNIT6  = '        '           / Physical units of field  6
TFORM7  = '1J      '           / FORTRAN format of field  7
TTYPE7  = 'FREQ ID         '   / Type (heading) of field  7
TUNIT7  = '        '           / Physical units of field  7
ISORTORD=                   1
END
```

B.4.11 AIPS OT tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   17 / Width of table in bytes
NAXIS2  =                20196 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    5 / Number of fields in each row
EXTNAME = 'AIPS OT '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1E      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'TIME INTERVAL   '   / Type (heading) of field  2
TUNIT2  = 'DAYS    '           / Physical units of field  2
```

```
TFORM3  = '1J      '            / FORTRAN format of field  3
TTYPE3  = 'SOURCE ID       '    / Type (heading) of field  3
TUNIT3  = '        '            / Physical units of field  3
TFORM4  = '1J      '            / FORTRAN format of field  4
TTYPE4  = 'ANTENNA         '    / Type (heading) of field  4
TUNIT4  = '        '            / Physical units of field  4
TFORM5  = '1L      '            / FORTRAN format of field  5
TTYPE5  = 'OVER TOP        '    / Type (heading) of field  5
TUNIT5  = '        '            / Physical units of field  5
REVISION=            1
END
```

B.4.12 AIPS PO tables

```
XTENSION= 'BINTABLE'            / Extension type
BITPIX  =                     8 / Binary data
NAXIS   =                     2 / Table is a matrix
NAXIS1  =                    36 / Width of table in bytes
NAXIS2  =                  2051 / Number of entries in table
PCOUNT  =                     0 / Random parameter count
GCOUNT  =                     1 / Group count
TFIELDS =                     5 / Number of fields in each row
EXTNAME = 'AIPS PO '            / AIPS table file
EXTVER  =                     1 / Version number of table
TFORM1  = '1D      '            / FORTRAN format of field  1
TTYPE1  = 'TIME            '    / Type (heading) of field  1
TUNIT1  = 'DAYS    '            / Physical units of field  1
TFORM2  = '1J      '            / FORTRAN format of field  2
TTYPE2  = 'SOURCE ID       '    / Type (heading) of field  2
TUNIT2  = '        '            / Physical units of field  2
TFORM3  = '1D      '            / FORTRAN format of field  3
TTYPE3  = 'SOURCE RA       '    / Type (heading) of field  3
TUNIT3  = 'DEGREES '            / Physical units of field  3
TFORM4  = '1D      '            / FORTRAN format of field  4
TTYPE4  = 'SOURCE DEC      '    / Type (heading) of field  4
TUNIT4  = 'DEGREES '            / Physical units of field  4
TFORM5  = '1D      '            / FORTRAN format of field  5
TTYPE5  = 'SOURCE DISTANCE '    / Type (heading) of field  5
TUNIT5  = 'AU      '            / Physical units of field  5
TABREV  =            1
RDATE   = '2006-05-07'
END
```

B.4.13 AIPS SU tables

```
XTENSION= 'BINTABLE'            / Extension type
BITPIX  =                     8 / Binary data
NAXIS   =                     2 / Table is a matrix
NAXIS1  =                   172 / Width of table in bytes
NAXIS2  =                     2 / Number of entries in table
PCOUNT  =                     0 / Random parameter count
GCOUNT  =                     1 / Group count
TFIELDS =                    19 / Number of fields in each row
EXTNAME = 'AIPS SU '            / AIPS table file
EXTVER  =                     1 / Version number of table
TFORM1  = '1J      '            / FORTRAN format of field  1
TTYPE1  = 'ID. NO.         '    / Type (heading) of field  1
```

```
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '16A     '           / FORTRAN format of field  2
TTYPE2  = 'SOURCE          '   / Type (heading) of field  2
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'QUAL            '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '4A      '           / FORTRAN format of field  4
TTYPE4  = 'CALCODE         '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '2E      '           / FORTRAN format of field  5
TTYPE5  = 'IFLUX           '   / Type (heading) of field  5
TUNIT5  = 'JY      '           / Physical units of field  5
TFORM6  = '2E      '           / FORTRAN format of field  6
TTYPE6  = 'QFLUX           '   / Type (heading) of field  6
TUNIT6  = 'JY      '           / Physical units of field  6
TFORM7  = '2E      '           / FORTRAN format of field  7
TTYPE7  = 'UFLUX           '   / Type (heading) of field  7
TUNIT7  = 'JY      '           / Physical units of field  7
TFORM8  = '2E      '           / FORTRAN format of field  8
TTYPE8  = 'VFLUX           '   / Type (heading) of field  8
TUNIT8  = 'JY      '           / Physical units of field  8
TFORM9  = '2D      '           / FORTRAN format of field  9
TTYPE9  = 'FREQOFF         '   / Type (heading) of field  9
TUNIT9  = 'HZ      '           / Physical units of field  9
TFORM10 = '1D      '           / FORTRAN format of field 10
TTYPE10 = 'BANDWIDTH       '   / Type (heading) of field 10
TUNIT10 = 'HZ      '           / physical units of field 10
TFORM11 = '1D      '           / FORTRAN format of field 11
TTYPE11 = 'RAEPO           '   / Type (heading) of field 11
TUNIT11 = 'DEGREES '           / physical units of field 11
TFORM12 = '1D      '           / FORTRAN format of field 12
TTYPE12 = 'DECEPO          '   / Type (heading) of field 12
TUNIT12 = 'DEGREES '           / physical units of field 12
TFORM13 = '1D      '           / FORTRAN format of field 13
TTYPE13 = 'EPOCH           '   / Type (heading) of field 13
TUNIT13 = 'YEARS   '           / physical units of field 13
TFORM14 = '1D      '           / FORTRAN format of field 14
TTYPE14 = 'RAAPP           '   / Type (heading) of field 14
TUNIT14 = 'DEGREES '           / physical units of field 14
TFORM15 = '1D      '           / FORTRAN format of field 15
TTYPE15 = 'DECAPP          '   / Type (heading) of field 15
TUNIT15 = 'DEGREES '           / physical units of field 15
TFORM16 = '2D      '           / FORTRAN format of field 16
TTYPE16 = 'LSRVEL          '   / Type (heading) of field 16
TUNIT16 = 'M/SEC   '           / physical units of field 16
TFORM17 = '2D      '           / FORTRAN format of field 17
TTYPE17 = 'RESTFREQ        '   / Type (heading) of field 17
TUNIT17 = 'HZ      '           / physical units of field 17
TFORM18 = '1D      '           / FORTRAN format of field 18
TTYPE18 = 'PMRA            '   / Type (heading) of field 18
TUNIT18 = 'DEG/DAY '           / physical units of field 18
TFORM19 = '1D      '           / FORTRAN format of field 19
TTYPE19 = 'PMDEC           '   / Type (heading) of field 19
TUNIT19 = 'DEG/DAY '           / physical units of field 19
NO_IF   =               2
FREQID  =               1
VELDEF  = '        '
VELTYP  = '        '
```

```
END
```

B.4.14 AIPS SY tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  416 / Width of table in bytes
NAXIS2  =                92406 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   12 / Number of fields in each row
EXTNAME = 'AIPS SY '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'TIME INTERVAL   '   / Type (heading) of field  2
TUNIT2  = 'DAYS    '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'CAL TYPE        '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'SOURCE ID       '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'ANTENNA NO.     '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1J      '           / FORTRAN format of field  6
TTYPE6  = 'SUBARRAY        '   / Type (heading) of field  6
TUNIT6  = '        '           / Physical units of field  6
TFORM7  = '1J      '           / FORTRAN format of field  7
TTYPE7  = 'FREQ ID         '   / Type (heading) of field  7
TUNIT7  = '        '           / Physical units of field  7
TFORM8  = '16E     '           / FORTRAN format of field  8
TTYPE8  = 'POWER DIF1      '   / Type (heading) of field  8
TUNIT8  = 'COUNTS  '           / Physical units of field  8
TFORM9  = '16E     '           / FORTRAN format of field  9
TTYPE9  = 'POWER SUM1      '   / Type (heading) of field  9
TUNIT9  = 'COUNTS  '           / Physical units of field  9
TFORM10 = '16E     '           / FORTRAN format of field 10
TTYPE10 = 'POST GAIN1      '   / Type (heading) of field 10
TUNIT11 = '        '           / Physical units of field 10
TFORM11 = '16E     '           / FORTRAN format of field 11
TTYPE11 = 'POWER DIF2      '   / Type (heading) of field 11
TUNIT11 = 'COUNTS  '           / physical units of field 11
TFORM12 = '16E     '           / FORTRAN format of field 12
TTYPE12 = 'POWER SUM2      '   / Type (heading) of field 12
TUNIT12 = 'COUNTS  '           / physical units of field 12
TFORM13 = '16E     '           / FORTRAN format of field 13
TTYPE13 = 'POST GAIN2      '   / Type (heading) of field 13
TUNIT13 = '        '           / physical units of field 13
NO_ANT  =          28
NO_POL  =           2
NO_IF   =          16
END
```

B.4.15 AIPS TY tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   40 / Width of table in bytes
NAXIS2  =                 1072 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   10 / Number of fields in each row
EXTNAME = 'AIPS TY '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1E      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'TIME INTERVAL   '   / Type (heading) of field  2
TUNIT2  = 'DAYS    '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'SOURCE ID       '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'ANTENNA NO.     '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'SUBARRAY        '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1J      '           / FORTRAN format of field  6
TTYPE6  = 'FREQ ID         '   / Type (heading) of field  6
TUNIT6  = '        '           / Physical units of field  6
TFORM7  = '1E      '           / FORTRAN format of field  7
TTYPE7  = 'TSYS 1          '   / Type (heading) of field  7
TUNIT7  = 'KELVINS '           / Physical units of field  7
TFORM8  = '1E      '           / FORTRAN format of field  8
TTYPE8  = 'TANT 1          '   / Type (heading) of field  8
TUNIT8  = 'KELVINS '           / Physical units of field  8
TFORM9  = '1E      '           / FORTRAN format of field  9
TTYPE9  = 'TSYS 2          '   / Type (heading) of field  9
TUNIT9  = 'KELVINS '           / Physical units of field  9
TFORM10 = '1E      '           / FORTRAN format of field 10
TTYPE10 = 'TANT 2          '   / Type (heading) of field 10
TUNIT10 = 'KELVINS '           / physical units of field 10
NO_POL  =            2
NO_IF   =            1
REVISION=           10
END
```

B.4.16 AIPS WX tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   56 / Width of table in bytes
NAXIS2  =                  602 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   13 / Number of fields in each row
EXTNAME = 'AIPS WX '           / AIPS table file
```

```
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '             / FORTRAN format of field  1
TTYPE1  = 'TIME            '     / Type (heading) of field  1
TUNIT1  = 'DAYS    '             / Physical units of field  1
TFORM2  = '1E      '             / FORTRAN format of field  2
TTYPE2  = 'TIME_INTERVAL   '     / Type (heading) of field  2
TUNIT2  = 'DAYS    '             / Physical units of field  2
TFORM3  = '1J      '             / FORTRAN format of field  3
TTYPE3  = 'ANTENNA_NO      '     / Type (heading) of field  3
TUNIT3  = '        '             / Physical units of field  3
TFORM4  = '1J      '             / FORTRAN format of field  4
TTYPE4  = 'SUBARRAY        '     / Type (heading) of field  4
TUNIT4  = '        '             / Physical units of field  4
TFORM5  = '1E      '             / FORTRAN format of field  5
TTYPE5  = 'TEMPERATURE     '     / Type (heading) of field  5
TUNIT5  = 'CENTIGRA'             / Physical units of field  5
TFORM6  = '1E      '             / FORTRAN format of field  6
TTYPE6  = 'PRESSURE        '     / Type (heading) of field  6
TUNIT6  = 'MILLIBAR'             / Physical units of field  6
TFORM7  = '1E      '             / FORTRAN format of field  7
TTYPE7  = 'DEWPOINT        '     / Type (heading) of field  7
TUNIT7  = 'CENTIGRA'             / Physical units of field  7
TFORM8  = '1E      '             / FORTRAN format of field  8
TTYPE8  = 'WIND_VELOCITY   '     / Type (heading) of field  8
TUNIT8  = 'M/SEC   '             / Physical units of field  8
TFORM9  = '1E      '             / FORTRAN format of field  9
TTYPE9  = 'WIND_DIRECTION  '     / Type (heading) of field  9
TUNIT9  = 'DEGREES '             / Physical units of field  9
TFORM10 = '1E      '             / FORTRAN format of field 10
TTYPE10 = 'WIND_GUST       '     / Type (heading) of field 10
TUNIT10 = 'M/SEC   '             / physical units of field 10
TFORM12 = '1E      '             / FORTRAN format of field 11
TTYPE12 = 'Preciptation    '     / Type (heading) of field 11
TUNIT12 = 'CM      '             / physical units of field 11
TFORM12 = '1E      '             / FORTRAN format of field 12
TTYPE12 = 'WVR_H2O         '     / Type (heading) of field 12
TUNIT12 = '/M/M    '             / physical units of field 12
TFORM13 = '1E      '             / FORTRAN format of field 13
TTYPE13 = 'IONOS_ELECTRON  '     / Type (heading) of field 13
TUNIT13 = '/M/M    '             / physical units of field 13
OBSCODE = 'AA000   '
RDATE   = '2011-09-26'
TABREV  =            3
END
```

B.5 AIPS calibration tables

B.5.1 AIPS BL tables

```
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   56 / Width of table in bytes
NAXIS2  =                  378 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   14 / Number of fields in each row
EXTNAME = 'AIPS BL '             / AIPS table file
EXTVER  =                    1 / Version number of table
```

```
TFORM1  = '1E      '          / FORTRAN format of field  1
TTYPE1  = 'TIME            '  / Type (heading) of field  1
TUNIT1  = 'DAYS    '          / Physical units of field  1
TFORM2  = '1J      '          / FORTRAN format of field  2
TTYPE2  = 'SOURCE ID       '  / Type (heading) of field  2
TUNIT2  = '        '          / Physical units of field  2
TFORM3  = '1J      '          / FORTRAN format of field  3
TTYPE3  = 'SUBARRAY        '  / Type (heading) of field  3
TUNIT3  = '        '          / Physical units of field  3
TFORM4  = '1J      '          / FORTRAN format of field  4
TTYPE4  = 'ANTENNA1        '  / Type (heading) of field  4
TUNIT4  = '        '          / Physical units of field  4
TFORM5  = '1J      '          / FORTRAN format of field  5
TTYPE5  = 'ANTENNA2        '  / Type (heading) of field  5
TUNIT5  = '        '          / Physical units of field  5
TFORM6  = '1J      '          / FORTRAN format of field  6
TTYPE6  = 'FREQ ID         '  / Type (heading) of field  6
TUNIT6  = '        '          / Physical units of field  6
TFORM7  = '1E      '          / FORTRAN format of field  7
TTYPE7  = 'REAL M1         '  / Type (heading) of field  7
TUNIT7  = '        '          / Physical units of field  7
TFORM8  = '1E      '          / FORTRAN format of field  8
TTYPE8  = 'IMAG M1         '  / Type (heading) of field  8
TUNIT8  = '        '          / Physical units of field  8
TFORM9  = '1E      '          / FORTRAN format of field  9
TTYPE9  = 'REAL A1         '  / Type (heading) of field  9
TUNIT9  = '        '          / Physical units of field  9
TFORM10 = '1E      '          / FORTRAN format of field 10
TTYPE10 = 'IMAG A1         '  / Type (heading) of field 10
TUNIT10 = '        '          / physical units of field 10
TFORM11 = '1E      '          / FORTRAN format of field 11
TTYPE11 = 'REAL M2         '  / Type (heading) of field 11
TUNIT11 = '        '          / physical units of field 11
TFORM12 = '1E      '          / FORTRAN format of field 12
TTYPE12 = 'IMAG M2         '  / Type (heading) of field 12
TUNIT12 = '        '          / physical units of field 12
TFORM13 = '1E      '          / FORTRAN format of field 13
TTYPE13 = 'REAL A2         '  / Type (heading) of field 13
TUNIT13 = '        '          / physical units of field 13
TFORM14 = '1E      '          / FORTRAN format of field 14
TTYPE14 = 'IMAG A2         '  / Type (heading) of field 14
TUNIT14 = '        '          / physical units of field 14
NO_ANT  =           28
NO_POL  =            2
NO_IF   =            1
END
```

B.5.2 AIPS BP tables

```
XTENSION= 'BINTABLE'          / Extension type
BITPIX  =                   8 / Binary data
NAXIS   =                   2 / Table is a matrix
NAXIS1  =                2120 / Width of table in bytes
NAXIS2  =                 224 / Number of entries in table
PCOUNT  =                   0 / Random parameter count
GCOUNT  =                   1 / Group count
TFIELDS =                  16 / Number of fields in each row
EXTNAME = 'AIPS BP '          / AIPS table file
```

```
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'INTERVAL        '   / Type (heading) of field  2
TUNIT2  = 'DAYS    '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'SOURCE ID       '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'SUBARRAY        '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'ANTENNA         '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1E      '           / FORTRAN format of field  6
TTYPE6  = 'BANDWIDTH       '   / Type (heading) of field  6
TUNIT6  = 'HZ      '           / Physical units of field  6
TFORM7  = '2D      '           / FORTRAN format of field  7
TTYPE7  = 'CHN_SHIFT       '   / Type (heading) of field  7
TUNIT7  = '        '           / Physical units of field  7
TFORM8  = '1J      '           / FORTRAN format of field  8
TTYPE8  = 'FREQ ID         '   / Type (heading) of field  8
TUNIT8  = '        '           / Physical units of field  8
TFORM9  = '1J      '           / FORTRAN format of field  9
TTYPE9  = 'REFANT 1        '   / Type (heading) of field  9
TUNIT9  = '        '           / Physical units of field  9
TFORM10 = '2E      '           / FORTRAN format of field 10
TTYPE10 = 'WEIGHT 1        '   / Type (heading) of field 10
TUNIT10 = '        '           / physical units of field 10
TFORM11 = '128E    '           / FORTRAN format of field 11
TTYPE11 = 'REAL 1          '   / Type (heading) of field 11
TUNIT11 = '        '           / physical units of field 11
TFORM12 = '128E    '           / FORTRAN format of field 12
TTYPE12 = 'IMAG 1          '   / Type (heading) of field 12
TUNIT12 = '        '           / physical units of field 12
TFORM13 = '1J      '           / FORTRAN format of field 13
TTYPE13 = 'REFANT 2        '   / Type (heading) of field 13
TUNIT13 = '        '           / physical units of field 13
TFORM14 = '2E      '           / FORTRAN format of field 14
TTYPE14 = 'WEIGHT 2        '   / Type (heading) of field 14
TUNIT14 = '        '           / physical units of field 14
TFORM15 = '128E    '           / FORTRAN format of field 15
TTYPE15 = 'REAL 2          '   / Type (heading) of field 15
TUNIT15 = '        '           / physical units of field 15
TFORM16 = '128E    '           / FORTRAN format of field 16
TTYPE16 = 'IMAG 2          '   / Type (heading) of field 16
TUNIT16 = '        '           / physical units of field 16
NO_ANT  =           28
NO_POL  =            2
NO_IF   =            2
NO_CHAN =           64
STRT_CHN=            1
NO_SHFTS=            1
LOW_SHFT=   0.0000000000000000E+00
SHFT_INC=   0.0000000000000000E+00
BP_TYPE = '        '
ISORTORD=                    5
```

```
END
```

B.5.3 AIPS CL tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  192 / Width of table in bytes
NAXIS2  =                 1225 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   33 / Number of fields in each row
EXTNAME = 'AIPS CL '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'TIME INTERVAL   '   / Type (heading) of field  2
TUNIT2  = 'DAYS    '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'SOURCE ID       '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'ANTENNA NO.     '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'SUBARRAY        '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1J      '           / FORTRAN format of field  6
TTYPE6  = 'FREQ ID         '   / Type (heading) of field  6
TUNIT6  = '        '           / Physical units of field  6
TFORM7  = '1E      '           / FORTRAN format of field  7
TTYPE7  = 'I.FAR.ROT       '   / Type (heading) of field  7
TUNIT7  = 'RAD/M**2'           / Physical units of field  7
TFORM8  = '1D      '           / FORTRAN format of field  8
TTYPE8  = 'GEODELAY        '   / Type (heading) of field  8
TUNIT8  = 'SECONDS '           / Physical units of field  8
TFORM9  = '2E      '           / FORTRAN format of field  9
TTYPE9  = 'DOPPOFF         '   / Type (heading) of field  9
TUNIT9  = 'HZ      '           / Physical units of field  9
TFORM10 = '1E      '           / FORTRAN format of field 10
TTYPE10 = 'ATMOS           '   / Type (heading) of field 10
TUNIT10 = 'SECONDS '           / physical units of field 10
TFORM11 = '1E      '           / FORTRAN format of field 11
TTYPE11 = 'DATMOS          '   / Type (heading) of field 11
TUNIT11 = 'SEC/SEC '           / physical units of field 11
TFORM12 = '1E      '           / FORTRAN format of field 12
TTYPE12 = 'MBDELAY1        '   / Type (heading) of field 12
TUNIT12 = 'SECONDS '           / physical units of field 12
TFORM13 = '1E      '           / FORTRAN format of field 13
TTYPE13 = 'CLOCK 1         '   / Type (heading) of field 13
TUNIT13 = 'SECONDS '           / physical units of field 13
TFORM14 = '1E      '           / FORTRAN format of field 14
TTYPE14 = 'DCLOCK 1        '   / Type (heading) of field 14
TUNIT14 = 'SEC/SEC '           / physical units of field 14
TFORM15 = '1E      '           / FORTRAN format of field 15
TTYPE15 = 'DISP 1          '   / Type (heading) of field 15
```

```
TUNIT15 = 'SEC/M**2'           / physical units of field 15
TFORM16 = '1E      '           / FORTRAN format of field 16
TTYPE16 = 'DDISP 1         '   / Type (heading) of field 16
TUNIT16 = 'S/S/M**2'           / physical units of field 16
TFORM17 = '2E      '           / FORTRAN format of field 17
TTYPE17 = 'REAL1           '   / Type (heading) of field 17
TUNIT17 = '        '           / physical units of field 17
TFORM18 = '2E      '           / FORTRAN format of field 18
TTYPE18 = 'IMAG1           '   / Type (heading) of field 18
TUNIT18 = '        '           / physical units of field 18
TFORM19 = '2E      '           / FORTRAN format of field 19
TTYPE19 = 'RATE 1          '   / Type (heading) of field 19
TUNIT19 = 'SEC/SEC '           / physical units of field 19
TFORM20 = '2E      '           / FORTRAN format of field 20
TTYPE20 = 'DELAY 1         '   / Type (heading) of field 20
TUNIT20 = 'SECONDS '           / physical units of field 20
TFORM21 = '2E      '           / FORTRAN format of field 21
TTYPE21 = 'WEIGHT 1        '   / Type (heading) of field 21
TUNIT21 = '        '           / physical units of field 21
TFORM22 = '2J      '           / FORTRAN format of field 22
TTYPE22 = 'REFANT 1        '   / Type (heading) of field 22
TUNIT22 = '        '           / physical units of field 22
TFORM23 = '1E      '           / FORTRAN format of field 23
TTYPE23 = 'MBDELAY2        '   / Type (heading) of field 23
TUNIT23 = 'SECONDS '           / physical units of field 23
TFORM24 = '1E      '           / FORTRAN format of field 24
TTYPE24 = 'CLOCK 2         '   / Type (heading) of field 24
TUNIT24 = 'SECONDS '           / physical units of field 24
TFORM25 = '1E      '           / FORTRAN format of field 25
TTYPE25 = 'DCLOCK 2        '   / Type (heading) of field 25
TUNIT25 = 'SEC/SEC '           / physical units of field 25
TFORM26 = '1E      '           / FORTRAN format of field 26
TTYPE26 = 'DISP 2          '   / Type (heading) of field 26
TUNIT26 = 'SEC/M**2'           / physical units of field 26
TFORM27 = '1E      '           / FORTRAN format of field 27
TTYPE27 = 'DDISP 2         '   / Type (heading) of field 27
TUNIT27 = 'S/S/M**2'           / physical units of field 27
TFORM28 = '2E      '           / FORTRAN format of field 28
TTYPE28 = 'REAL2           '   / Type (heading) of field 28
TUNIT28 = '        '           / physical units of field 28
TFORM29 = '2E      '           / FORTRAN format of field 29
TTYPE29 = 'IMAG2           '   / Type (heading) of field 29
TUNIT29 = '        '           / physical units of field 29
TFORM30 = '2E      '           / FORTRAN format of field 30
TTYPE30 = 'RATE 2          '   / Type (heading) of field 30
TUNIT30 = 'SEC/SEC '           / physical units of field 30
TFORM31 = '2E      '           / FORTRAN format of field 31
TTYPE31 = 'DELAY 2         '   / Type (heading) of field 31
TUNIT31 = 'SECONDS '           / physical units of field 31
TFORM32 = '2E      '           / FORTRAN format of field 32
TTYPE32 = 'WEIGHT 2        '   / Type (heading) of field 32
TUNIT32 = '        '           / physical units of field 32
TFORM33 = '2J      '           / FORTRAN format of field 33
TTYPE33 = 'REFANT 2        '   / Type (heading) of field 33
TUNIT33 = '        '           / physical units of field 33
NO_ANT  =           28
NO_POL  =            2
NO_IF   =            2
NO_TERM =            1
```

```
MGMOD   =   0.10000000000000000E+01
REVISION=           10
ISORTORD=                  4
END
```

B.5.4 AIPS CP tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                16404 / Width of table in bytes
NAXIS2  =                    1 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    6 / Number of fields in each row
EXTNAME = 'AIPS CP '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '16A     '           / FORTRAN format of field  1
TTYPE1  = 'SOURCE          '   / Type (heading) of field  1
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '1J      '           / FORTRAN format of field  2
TTYPE2  = 'SOURCE ID       '   / Type (heading) of field  2
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '1024E   '           / FORTRAN format of field  3
TTYPE3  = 'I               '   / Type (heading) of field  3
TUNIT3  = 'Jy      '           / Physical units of field  3
TFORM4  = '1024E   '           / FORTRAN format of field  4
TTYPE4  = 'Q               '   / Type (heading) of field  4
TUNIT4  = 'Jy      '           / Physical units of field  4
TFORM5  = '1024E   '           / FORTRAN format of field  5
TTYPE5  = 'U               '   / Type (heading) of field  5
TUNIT5  = 'Jy      '           / Physical units of field  5
TFORM6  = '1024E   '           / FORTRAN format of field  6
TTYPE6  = 'V               '   / Type (heading) of field  6
TUNIT6  = 'Jy      '           / Physical units of field  6
NO_IF   =           16
NO_CHAN =           64
FREQID  =            1
END
```

B.5.5 AIPS PD tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                16400 / Width of table in bytes
NAXIS2  =                   28 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    9 / Number of fields in each row
EXTNAME = 'AIPS PD '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1J      '           / FORTRAN format of field  1
TTYPE1  = 'ANTENNA         '   / Type (heading) of field  1
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '1J      '           / FORTRAN format of field  2
TTYPE2  = 'SUBARRAY        '   / Type (heading) of field  2
```

```
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'FREQ ID         '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'REFANT          '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '0E      '           / FORTRAN format of field  5
TTYPE5  = 'P_DIFF          '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1024E   '           / FORTRAN format of field  6
TTYPE6  = 'REAL 1          '   / Type (heading) of field  6
TUNIT6  = '        '           / Physical units of field  6
TFORM7  = '1024E   '           / FORTRAN format of field  7
TTYPE7  = 'IMAG 1          '   / Type (heading) of field  7
TUNIT7  = '        '           / Physical units of field  7
TFORM8  = '1024E   '           / FORTRAN format of field  8
TTYPE8  = 'REAL 2          '   / Type (heading) of field  8
TUNIT8  = '        '           / Physical units of field  8
TFORM9  = '1024E   '           / FORTRAN format of field  9
TTYPE9  = 'IMAG 2          '   / Type (heading) of field  9
TUNIT9  = '        '           / Physical units of field  9
NO_ANT  =           28
NO_POL  =            2
NO_IF   =           16
NO_CHAN =           64
POLTYPE = 'APPROX  '
END
```

B.5.6 AIPS PP tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  516 / Width of table in bytes
NAXIS2  =                    1 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    3 / Number of fields in each row
EXTNAME = 'AIPS PP '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1J      '           / FORTRAN format of field  1
TTYPE1  = 'SUBARRAY        '   / Type (heading) of field  1
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '1J      '           / FORTRAN format of field  2
TTYPE2  = 'FREQ ID         '   / Type (heading) of field  2
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '127E    '           / FORTRAN format of field  3
TTYPE3  = 'PHASE DIFF      '   / Type (heading) of field  3
TUNIT3  = 'DEGREES '           / Physical units of field  3
NO_IF   =            1
NO_CHAN =          127
BIF     =            1
BCHAN   =            1
END
```

B.5.7 AIPS SN tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  140 / Width of table in bytes
NAXIS2  =                   24 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   22 / Number of fields in each row
EXTNAME = 'AIPS SN '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '           / FORTRAN format of field  1
TTYPE1  = 'TIME            '   / Type (heading) of field  1
TUNIT1  = 'DAYS    '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'TIME INTERVAL   '   / Type (heading) of field  2
TUNIT2  = 'DAYS    '           / Physical units of field  2
TFORM3  = '1J      '           / FORTRAN format of field  3
TTYPE3  = 'SOURCE ID       '   / Type (heading) of field  3
TUNIT3  = '        '           / Physical units of field  3
TFORM4  = '1J      '           / FORTRAN format of field  4
TTYPE4  = 'ANTENNA NO.     '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '1J      '           / FORTRAN format of field  5
TTYPE5  = 'SUBARRAY        '   / Type (heading) of field  5
TUNIT5  = '        '           / Physical units of field  5
TFORM6  = '1J      '           / FORTRAN format of field  6
TTYPE6  = 'FREQ ID         '   / Type (heading) of field  6
TUNIT6  = '        '           / Physical units of field  6
TFORM7  = '1E      '           / FORTRAN format of field  7
TTYPE7  = 'I.FAR.ROT       '   / Type (heading) of field  7
TUNIT7  = 'RAD/M**2'           / Physical units of field  7
TFORM8  = '1J      '           / FORTRAN format of field  8
TTYPE8  = 'NODE NO.        '   / Type (heading) of field  8
TUNIT8  = '        '           / Physical units of field  8
TFORM9  = '1E      '           / FORTRAN format of field  9
TTYPE9  = 'MBDELAY1        '   / Type (heading) of field  9
TUNIT9  = 'SECONDS '           / Physical units of field  9
TFORM10 = '2E      '           / FORTRAN format of field 10
TTYPE10 = 'REAL1           '   / Type (heading) of field 10
TUNIT10 = '        '           / physical units of field 10
TFORM11 = '2E      '           / FORTRAN format of field 11
TTYPE11 = 'IMAG1           '   / Type (heading) of field 11
TUNIT11 = '        '           / physical units of field 11
TFORM12 = '2E      '           / FORTRAN format of field 12
TTYPE12 = 'DELAY 1         '   / Type (heading) of field 12
TUNIT12 = 'SECONDS '           / physical units of field 12
TFORM13 = '2E      '           / FORTRAN format of field 13
TTYPE13 = 'RATE 1          '   / Type (heading) of field 13
TUNIT13 = 'SEC/SEC '           / physical units of field 13
TFORM14 = '2E      '           / FORTRAN format of field 14
TTYPE14 = 'WEIGHT 1        '   / Type (heading) of field 14
TUNIT14 = '        '           / physical units of field 14
TFORM15 = '2J      '           / FORTRAN format of field 15
TTYPE15 = 'REFANT 1        '   / Type (heading) of field 15
TUNIT15 = '        '           / physical units of field 15
TFORM16 = '1E      '           / FORTRAN format of field 16
TTYPE16 = 'MBDELAY2        '   / Type (heading) of field 16
```

```
TUNIT16 = 'SECONDS '           / physical units of field 16
TFORM17 = '2E      '           / FORTRAN format of field 17
TTYPE17 = 'REAL2           '   / Type (heading) of field 17
TUNIT17 = '        '           / physical units of field 17
TFORM18 = '2E      '           / FORTRAN format of field 18
TTYPE18 = 'IMAG2           '   / Type (heading) of field 18
TUNIT18 = '        '           / physical units of field 18
TFORM19 = '2E      '           / FORTRAN format of field 19
TTYPE19 = 'DELAY 2         '   / Type (heading) of field 19
TUNIT19 = 'SECONDS '           / physical units of field 19
TFORM20 = '2E      '           / FORTRAN format of field 20
TTYPE20 = 'RATE 2          '   / Type (heading) of field 20
TUNIT20 = 'SEC/SEC '           / physical units of field 20
TFORM21 = '2E      '           / FORTRAN format of field 21
TTYPE21 = 'WEIGHT 2        '   / Type (heading) of field 21
TUNIT21 = '        '           / physical units of field 21
TFORM22 = '2J      '           / FORTRAN format of field 22
TTYPE22 = 'REFANT 2        '   / Type (heading) of field 22
TUNIT22 = '        '           / physical units of field 22
NO_ANT  =           28
NO_POL  =            2
NO_IF   =            2
NO_NODES=            0
MGMOD   =   0.10000000000000000E+01
APPLIED =                    T
REVISION=           10
SNORIGIN=            0
ISORTORD=                   1
END
```

B.6 UV-table HDU of *uv* table form

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                 6176 / Width of table in bytes
NAXIS2  =               609900 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    9 / Number of fields in each row
EXTNAME = 'AIPS UV '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1E      '           / FORTRAN format of field  1
TTYPE1  = 'UU--            '   / Type (heading) of field  1
TUNIT1  = 'SECONDS '           / physical units of field  1
TSCAL1  =  2.0424836601307E-10 / scale to physical units in field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'VV--            '   / Type (heading) of field  2
TUNIT2  = 'SECONDS '           / physical units of field  2
TSCAL2  =  2.0424836601307E-10 / scale to physical units in field  2
TFORM3  = '1E      '           / FORTRAN format of field  3
TTYPE3  = 'WW--            '   / Type (heading) of field  3
TUNIT3  = 'SECONDS '           / physical units of field  3
TSCAL3  =  2.0424836601307E-10 / scale to physical units in field  3
TFORM4  = '1E      '           / FORTRAN format of field  4
TTYPE4  = 'DATE            '   / Type (heading) of field  4
TUNIT4  = 'DAYS    '           / physical units of field  4
TZERO4  =  2.4552635000000E+06 / offset to physical units in field  4
```

```
TFORM5  = '1E      '           / FORTRAN format of field  5
TTYPE5  = 'BASELINE        '   / Type (heading) of field  5
TUNIT5  = '        '           / physical units of field  5
TFORM6  = '1E      '           / FORTRAN format of field  6
TTYPE6  = 'FREQSEL         '   / Type (heading) of field  6
TUNIT6  = '        '           / physical units of field  6
TFORM7  = '1E      '           / FORTRAN format of field  7
TTYPE7  = 'SOURCE          '   / Type (heading) of field  7
TUNIT7  = '        '           / physical units of field  7
TFORM8  = '1E      '           / FORTRAN format of field  8
TTYPE8  = 'INTTIM          '   / Type (heading) of field  8
TUNIT8  = '        '           / physical units of field  8
TFORM9  = '1536E   '           / FORTRAN format of field  9
TTYPE9  = 'VISIBILITIES    '   / Type (heading) of field  9
TUNIT9  = 'JY      '           / physical units of field  9
TDIM9   = '(3,4,64,2,1,1)'

1CTYP9  = 'COMPLEX '           / 1=real,2=imag,3=weight
1CRVL9  =    1.00000000000E+00 /
1CDLT9  =      1.000000000E+00 /
1CRPX9  =      1.000000000E+00 /
1CROT9  =      0.000000000E+00 /
2CTYP9  = 'STOKES  '           / -1=RR, -2=LL, -3=RL, -4=LR
2CRVL9  =   -1.00000000000E+00 /
2CDLT9  =     -1.000000000E+00 /
2CRPX9  =      1.000000000E+00 /
2CROT9  =      0.000000000E+00 /
3CTYP9  = 'FREQ    '           / Frequency in Hz.
3CRVL9  =    4.89600000000E+09 /
3CDLT9  =      2.000000000E+06 /
3CRPX9  =      3.300000000E+01 /
3CROT9  =      0.000000000E+00 /
4CTYP9  = 'IF      '           / Freq. group no. in CH table
4CRVL9  =    1.00000000000E+00 /
4CDLT9  =      1.000000000E+00 /
4CRPX9  =      1.000000000E+00 /
4CROT9  =      0.000000000E+00 /
5CTYP9  = 'RA      '           / Right Ascension in deg.
5CRVL9  =    0.00000000000E+00 /
5CDLT9  =      1.000000000E+00 /
5CRPX9  =      1.000000000E+00 /
5CROT9  =      0.000000000E+00 /
6CTYP9  = 'DEC     '           / Declination in deg.
6CRVL9  =    0.00000000000E+00 /
6CDLT9  =      1.000000000E+00 /
6CRPX9  =      1.000000000E+00 /
6CROT9  =      0.000000000E+00 /
OBJECT  = 'MULTI   '           / Source name
TELESCOP= 'EVLA    '           /
INSTRUME= 'EVLA    '           /
OBSERVER= 'TOSR0001'           /
DATE-OBS= '2010-03-08'         / Obs start date YYYY-MM-DD
DATE-MAP= '2010-03-08'         / Last processing date YYYY-MM-DD
BSCALE  =    1.00000000000E+00 / REAL = TAPE * BSCALE + BZERO
BZERO   =    0.00000000000E+00 /
BUNIT   = 'UNCALIB '           / Units of flux
EQUINOX =      2.000000000E+03 / Epoch of RA DEC
COMMENT / IEEE not-a-number used for blanked f.p. pixels
ALTRPIX =      3.300000000E+01 / Altenate FREQ/VEL ref pixel
```

```
HISTORY AIPS   SORT ORDER = 'TB'
              / Where T means TIME (IAT)
              / Where B means BASELINE NUM
HISTORY AIPS   IMNAME='Empty field ' IMCLASS='UVDATA' IMSEQ=   1     /
HISTORY AIPS   USERNO= 2010           /
HISTORY AIPS   IPIECE= 1 NPIECE= 1     / piece number
HISTORY AIPS   FIRSTVIS=          1     / first vis #
HISTORY AIPS   LASTVIS =      609900     / last vis #
END
```

B.7 UV-table HDU of *uv* table form, compressed data

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                 2088 / Width of table in bytes
NAXIS2  =               609900 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   11 / Number of fields in each row
EXTNAME = 'AIPS UV '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1E      '           / FORTRAN format of field  1
TTYPE1  = 'UU--            '   / Type (heading) of field  1
TUNIT1  = 'SECONDS '           / physical units of field  1
TSCAL1  =  2.0424836601307E-10 / scale to physical units in field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'VV--            '   / Type (heading) of field  2
TUNIT2  = 'SECONDS '           / physical units of field  2
TSCAL2  =  2.0424836601307E-10 / scale to physical units in field  2
TFORM3  = '1E      '           / FORTRAN format of field  3
TTYPE3  = 'WW--            '   / Type (heading) of field  3
TUNIT3  = 'SECONDS '           / physical units of field  3
TSCAL3  =  2.0424836601307E-10 / scale to physical units in field  3
TFORM4  = '1E      '           / FORTRAN format of field  4
TTYPE4  = 'DATE            '   / Type (heading) of field  4
TUNIT4  = 'DAYS    '           / physical units of field  4
TZERO4  =  2.4552635000000E+06 / offset to physical units in field  4
TFORM5  = '1E      '           / FORTRAN format of field  5
TTYPE5  = 'BASELINE        '   / Type (heading) of field  5
TUNIT5  = '        '           / physical units of field  5
TFORM6  = '1E      '           / FORTRAN format of field  6
TTYPE6  = 'FREQSEL         '   / Type (heading) of field  6
TUNIT6  = '        '           / physical units of field  6
TFORM7  = '1E      '           / FORTRAN format of field  7
TTYPE7  = 'SOURCE          '   / Type (heading) of field  7
TUNIT7  = '        '           / physical units of field  7
TFORM8  = '1E      '           / FORTRAN format of field  8
TTYPE8  = 'INTTIM          '   / Type (heading) of field  8
TUNIT8  = '        '           / physical units of field  8
TFORM9  = '1E      '           / FORTRAN format of field  9
TTYPE9  = 'WEIGHT          '   / Type (heading) of field  9
TUNIT9  = '        '           / physical units of field  9
TFORM10 = '1E      '           / FORTRAN format of field 10
TTYPE10 = 'SCALE           '   / Type (heading) of field 10
TUNIT10 = '        '           / physical units of field 10
TFORM11 = '1024I   '           / FORTRAN format of field 11
TTYPE11 = 'VISIBILITIES    '   / Type (heading) of field 11
```

```
TUNIT11 = 'JY      '            / physical units of field 11
TNULL11 =               -32767 / magic value for flagged data
TDIM11  = '(2,4,64,2,1,1)'

1CTYP11 = 'COMPLEX '           / 1=real,2=imag,3=weight
1CRVL11 =    1.00000000000E+00 /
1CDLT11 =      1.000000000E+00 /
1CRPX11 =      1.000000000E+00 /
1CROT11 =      0.000000000E+00 /
2CTYP11 = 'STOKES  '           / -1=RR, -2=LL, -3=RL, -4=LR
2CRVL11 =   -1.00000000000E+00 /
2CDLT11 =     -1.000000000E+00 /
2CRPX11 =      1.000000000E+00 /
2CROT11 =      0.000000000E+00 /
3CTYP11 = 'FREQ    '           / Frequency in Hz.
3CRVL11 =    4.89600000000E+09 /
3CDLT11 =      2.000000000E+06 /
3CRPX11 =      3.300000000E+01 /
3CROT11 =      0.000000000E+00 /
4CTYP11 = 'IF      '           / Freq. group no. in CH table
4CRVL11 =    1.00000000000E+00 /
4CDLT11 =      1.000000000E+00 /
4CRPX11 =      1.000000000E+00 /
4CROT11 =      0.000000000E+00 /
5CTYP11 = 'RA      '           / Right Ascension in deg.
5CRVL11 =    0.00000000000E+00 /
5CDLT11 =      1.000000000E+00 /
5CRPX11 =      1.000000000E+00 /
5CROT11 =      0.000000000E+00 /
6CTYP11 = 'DEC     '           / Declination in deg.
6CRVL11 =    0.00000000000E+00 /
6CDLT11 =      1.000000000E+00 /
6CRPX11 =      1.000000000E+00 /
6CROT11 =      0.000000000E+00 /
OBJECT  = 'MULTI   '           / Source name
TELESCOP= 'EVLA    '           /
INSTRUME= 'EVLA    '           /
OBSERVER= 'TOSR0001'           /
DATE-OBS= '2010-03-08'         / Obs start date YYYY-MM-DD
DATE-MAP= '2012-01-06'         / Last processing date YYYY-MM-DD
BSCALE  =    1.00000000000E+00 / REAL = TAPE * BSCALE + BZERO
BZERO   =    0.00000000000E+00 /
BUNIT   = 'UNCALIB '           / Units of flux
EQUINOX =      2.000000000E+03 / Epoch of RA DEC
COMMENT / IEEE not-a-number used for blanked f.p. pixels
ALTRPIX =      3.300000000E+01 / Altenate FREQ/VEL ref pixel
HISTORY AIPS   SORT ORDER = 'TB'
              / Where T means TIME (IAT)
              / Where B means BASELINE NUM
HISTORY AIPS   IMNAME='Memo 117    ' IMCLASS='UVCMP ' IMSEQ=   1     /
HISTORY AIPS   USERNO= 2010            /
HISTORY AIPS   IPIECE= 1 NPIECE= 1     / piece number
HISTORY AIPS   FIRSTVIS=           1     / first vis #
HISTORY AIPS   LASTVIS =      609900     / last vis #
END
```

B.8 AIPS tables primarily related to imaging

B.8.1 AIPS CC tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   12 / Width of table in bytes
NAXIS2  =                  500 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    3 / Number of fields in each row
EXTNAME = 'AIPS CC '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1E      '           / FORTRAN format of field  1
TTYPE1  = 'FLUX            '   / Type (heading) of field  1
TUNIT1  = 'JY      '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'DELTAX          '   / Type (heading) of field  2
TUNIT2  = 'DEGREES '           / Physical units of field  2
TFORM3  = '1E      '           / FORTRAN format of field  3
TTYPE3  = 'DELTAY          '   / Type (heading) of field  3
TUNIT3  = 'DEGREES '           / Physical units of field  3
TFORM4  = '1E      '           / FORTRAN format of field  4
TTYPE4  = 'DELTAZ          '   / Type (heading) of field  4
TUNIT4  = 'DEGREES '           / Physical units of field  4
TFORM5  = '1E      '           / FORTRAN format of field  4
TTYPE5  = 'MAJOR AX        '   / Type (heading) of field  4
TUNIT5  = 'DEGREES '           / Physical units of field  4
TFORM6  = '1E      '           / FORTRAN format of field  4
TTYPE6  = 'MINOR AX        '   / Type (heading) of field  4
TUNIT6  = 'DEGREES '           / Physical units of field  4
TFORM7  = '1E      '           / FORTRAN format of field  4
TTYPE7  = 'POSANGLE        '   / Type (heading) of field  4
TUNIT7  = 'DEGREES '           / Physical units of field  4
TFORM8  = '1E      '           / FORTRAN format of field  4
TTYPE8  = 'TYPE OBJ        '   / Type (heading) of field  4
TUNIT8  = '        '           / Physical units of field  4
END
```

B.8.2 AIPS CG tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                   20 / Width of table in bytes
NAXIS2  =                  961 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                    4 / Number of fields in each row
EXTNAME = 'AIPS CG '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1D      '           / FORTRAN format of field  1
TTYPE1  = 'FREQUENCY       '   / Type (heading) of field  1
TUNIT1  = 'HZ      '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'BMAJ            '   / Type (heading) of field  2
```

```
TUNIT2  = 'DEGREES '           / Physical units of field  2
TFORM3  = '1E      '           / FORTRAN format of field  3
TTYPE3  = 'BMIN            '   / Type (heading) of field  3
TUNIT3  = 'DEGREES '           / Physical units of field  3
TFORM4  = '1E      '           / FORTRAN format of field  4
TTYPE4  = 'BPA             '   / Type (heading) of field  4
TUNIT4  = 'DEGREES '           / Physical units of field  4
TABREV  =            1
END
```

B.8.3 AIPS MF tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  164 / Width of table in bytes
NAXIS2  =                    1 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   41 / Number of fields in each row
EXTNAME = 'AIPS MF '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '1E      '           / FORTRAN format of field  1
TTYPE1  = 'PLANE           '   / Type (heading) of field  1
TUNIT1  = '        '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'PEAK INT        '   / Type (heading) of field  2
TUNIT2  = 'JY/BEAM '           / Physical units of field  2
TFORM3  = '1E      '           / FORTRAN format of field  3
TTYPE3  = 'I FLUX          '   / Type (heading) of field  3
TUNIT3  = 'JY      '           / Physical units of field  3
TFORM4  = '1E      '           / FORTRAN format of field  4
TTYPE4  = 'DELTAX          '   / Type (heading) of field  4
TUNIT4  = 'DEGREES '           / Physical units of field  4
TFORM5  = '1E      '           / FORTRAN format of field  5
TTYPE5  = 'DELTAY          '   / Type (heading) of field  5
TUNIT5  = 'DEGREES '           / Physical units of field  5
TFORM6  = '1E      '           / FORTRAN format of field  6
TTYPE6  = 'MAJOR AX        '   / Type (heading) of field  6
TUNIT6  = 'DEGREES '           / Physical units of field  6
TFORM7  = '1E      '           / FORTRAN format of field  7
TTYPE7  = 'MINOR AX        '   / Type (heading) of field  7
TUNIT7  = 'DEGREES '           / Physical units of field  7
TFORM8  = '1E      '           / FORTRAN format of field  8
TTYPE8  = 'POSANGLE        '   / Type (heading) of field  8
TUNIT8  = 'DEGREES '           / Physical units of field  8
TFORM9  = '1E      '           / FORTRAN format of field  9
TTYPE9  = 'Q FLUX          '   / Type (heading) of field  9
TUNIT9  = 'JY      '           / Physical units of field  9
TFORM10 = '1E      '           / FORTRAN format of field 10
TTYPE10 = 'U FLUX          '   / Type (heading) of field 10
TUNIT10 = 'JY      '           / physical units of field 10
TFORM11 = '1E      '           / FORTRAN format of field 11
TTYPE11 = 'V FLUX          '   / Type (heading) of field 11
TUNIT11 = 'JY      '           / physical units of field 11
TFORM12 = '1E      '           / FORTRAN format of field 12
TTYPE12 = 'ERR PEAK        '   / Type (heading) of field 12
TUNIT12 = 'JY/BEAM '           / physical units of field 12
```

```
TFORM13 = '1E      '          / FORTRAN format of field 13
TTYPE13 = 'ERR FLUX        '  / Type (heading) of field 13
TUNIT13 = 'JY      '          / physical units of field 13
TFORM14 = '1E      '          / FORTRAN format of field 14
TTYPE14 = 'ERR DLTX        '  / Type (heading) of field 14
TUNIT14 = 'DEGREES '          / physical units of field 14
TFORM15 = '1E      '          / FORTRAN format of field 15
TTYPE15 = 'ERR DLTY        '  / Type (heading) of field 15
TUNIT15 = 'DEGREES '          / physical units of field 15
TFORM16 = '1E      '          / FORTRAN format of field 16
TTYPE16 = 'ERR MAJA        '  / Type (heading) of field 16
TUNIT16 = 'DEGREES '          / physical units of field 16
TFORM17 = '1E      '          / FORTRAN format of field 17
TTYPE17 = 'ERR MINA        '  / Type (heading) of field 17
TUNIT17 = 'DEGREES '          / physical units of field 17
TFORM18 = '1E      '          / FORTRAN format of field 18
TTYPE18 = 'ERR PA          '  / Type (heading) of field 18
TUNIT18 = 'DEGREES '          / physical units of field 18
TFORM19 = '1E      '          / FORTRAN format of field 19
TTYPE19 = 'ERR QFLX        '  / Type (heading) of field 19
TUNIT19 = 'JY      '          / physical units of field 19
TFORM20 = '1E      '          / FORTRAN format of field 20
TTYPE20 = 'ERR UFLX        '  / Type (heading) of field 20
TUNIT20 = 'JY      '          / physical units of field 20
TFORM21 = '1E      '          / FORTRAN format of field 21
TTYPE21 = 'ERR VFLX        '  / Type (heading) of field 21
TUNIT21 = 'JY      '          / physical units of field 21
TFORM22 = '1E      '          / FORTRAN format of field 22
TTYPE22 = 'TYPE MOD        '  / Type (heading) of field 22
TUNIT22 = '        '          / physical units of field 22
TFORM23 = '1E      '          / FORTRAN format of field 23
TTYPE23 = 'D0 MAJOR        '  / Type (heading) of field 23
TUNIT23 = 'DEGREES '          / physical units of field 23
TFORM24 = '1E      '          / FORTRAN format of field 24
TTYPE24 = 'D0 MINOR        '  / Type (heading) of field 24
TUNIT24 = 'DEGREES '          / physical units of field 24
TFORM25 = '1E      '          / FORTRAN format of field 25
TTYPE25 = 'D0 POSAN        '  / Type (heading) of field 25
TUNIT25 = 'DEGREES '          / physical units of field 25
TFORM26 = '1E      '          / FORTRAN format of field 26
TTYPE26 = 'D- MAJOR        '  / Type (heading) of field 26
TUNIT26 = 'DEGREES '          / physical units of field 26
TFORM27 = '1E      '          / FORTRAN format of field 27
TTYPE27 = 'D- MINOR        '  / Type (heading) of field 27
TUNIT27 = 'DEGREES '          / physical units of field 27
TFORM28 = '1E      '          / FORTRAN format of field 28
TTYPE28 = 'D- POSAN        '  / Type (heading) of field 28
TUNIT28 = 'DEGREES '          / physical units of field 28
TFORM29 = '1E      '          / FORTRAN format of field 29
TTYPE29 = 'D+ MAJOR        '  / Type (heading) of field 29
TUNIT29 = 'DEGREES '          / physical units of field 29
TFORM30 = '1E      '          / FORTRAN format of field 30
TTYPE30 = 'D+ MINOR        '  / Type (heading) of field 30
TUNIT30 = 'DEGREES '          / physical units of field 30
TFORM31 = '1E      '          / FORTRAN format of field 31
TTYPE31 = 'D+ POSAN        '  / Type (heading) of field 31
TUNIT31 = 'DEGREES '          / physical units of field 31
TFORM32 = '1E      '          / FORTRAN format of field 32
TTYPE32 = 'RES RMS         '  / Type (heading) of field 32
```

```
TUNIT32 = 'JY/BEAM '           / physical units of field 32
TFORM33 = '1E      '           / FORTRAN format of field 33
TTYPE33 = 'RES PEAK        '   / Type (heading) of field 33
TUNIT33 = 'JY/BEAM '           / physical units of field 33
TFORM34 = '1E      '           / FORTRAN format of field 34
TTYPE34 = 'RES FLUX        '   / Type (heading) of field 34
TUNIT34 = 'JY      '           / physical units of field 34
TFORM35 = '1E      '           / FORTRAN format of field 35
TTYPE35 = 'CENTER X        '   / Type (heading) of field 35
TUNIT35 = 'PIXELS  '           / physical units of field 35
TFORM36 = '1E      '           / FORTRAN format of field 36
TTYPE36 = 'CENTER Y        '   / Type (heading) of field 36
TUNIT36 = 'PIXELS  '           / physical units of field 36
TFORM37 = '1E      '           / FORTRAN format of field 37
TTYPE37 = 'MAJ AXIS        '   / Type (heading) of field 37
TUNIT37 = 'PIXELS  '           / physical units of field 37
TFORM38 = '1E      '           / FORTRAN format of field 38
TTYPE38 = 'MIN AXIS        '   / Type (heading) of field 38
TUNIT38 = 'PIXELS  '           / physical units of field 38
TFORM39 = '1E      '           / FORTRAN format of field 39
TTYPE39 = 'PIXEL PA        '   / Type (heading) of field 39
TUNIT39 = 'DEGREES '           / physical units of field 39
TFORM40 = '1E      '           / FORTRAN format of field 40
TTYPE40 = 'PB FACT         '   / Type (heading) of field 40
TUNIT40 = '        '           / physical units of field 40
TFORM41 = '1E      '           / FORTRAN format of field 41
TTYPE41 = 'DLY FACT        '   / Type (heading) of field 41
TUNIT41 = '        '           / physical units of field 41
REVISION=            4
DEPTH1  =           45
DEPTH2  =            1
DEPTH3  =            1
DEPTH4  =            1
DEPTH5  =            1
REALRMS =   0.3615742E-02
END
```

B.8.4 AIPS RM tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  164 / Width of table in bytes
NAXIS2  =                16384 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   14 / Number of fields in each row
EXTNAME = 'AIPS RM '           / AIPS table file
EXTVER  =                    1 / Version number of table
TFORM1  = '2J      '           / FORTRAN format of field  1
TTYPE1  = 'PIXEL           '   / Type (heading) of field  1
TUNIT1  = 'PIXELS  '           / Physical units of field  1
TFORM2  = '1J      '           / FORTRAN format of field  2
TTYPE2  = 'NCOMP           '   / Type (heading) of field  2
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '1E      '           / FORTRAN format of field  3
TTYPE3  = 'IFLUX           '   / Type (heading) of field  3
```

```
TUNIT3  = 'JY/BEAM '          / Physical units of field  3
TFORM4  = '1E      '          / FORTRAN format of field  4
TTYPE4  = 'PFLUX           '  / Type (heading) of field  4
TUNIT4  = 'JY/BEAM '          / Physical units of field  4
TFORM5  = '4E      '          / FORTRAN format of field  5
TTYPE5  = 'PONE            '  / Type (heading) of field  5
TUNIT5  = 'JY/BEAM '          / Physical units of field  5
TFORM6  = '4E      '          / FORTRAN format of field  6
TTYPE6  = 'THETA           '  / Type (heading) of field  6
TUNIT6  = 'DEGREES '          / Physical units of field  6
TFORM7  = '4E      '          / FORTRAN format of field  7
TTYPE7  = 'ROT.MEAS        '  / Type (heading) of field  7
TUNIT7  = 'RAD/M/M '          / Physical units of field  7
TFORM8  = '4E      '          / FORTRAN format of field  8
TTYPE8  = 'THICKNESS       '  / Type (heading) of field  8
TUNIT8  = 'RAD/M/M '          / Physical units of field  8
TFORM9  = '4E      '          / FORTRAN format of field  9
TTYPE9  = 'ERR P1          '  / Type (heading) of field  9
TUNIT9  = 'JY/BEAM '          / Physical units of field  9
TFORM10 = '4E      '          / FORTRAN format of field 10
TTYPE10 = 'ERR THET        '  / Type (heading) of field 10RM
TUNIT10 = 'DEGREES '          / physical units of field 10
TFORM11 = '4E      '          / FORTRAN format of field 11
TTYPE11 = 'ERR RM          '  / Type (heading) of field 11
TUNIT11 = 'RAD/M/M '          / physical units of field 11
TFORM12 = '4E      '          / FORTRAN format of field 12
TTYPE12 = 'ERR THICK       '  / Type (heading) of field 12
TUNIT12 = 'RAD/M/M '          / physical units of field 12
TFORM13 = '2E      '          / FORTRAN format of field 13
TTYPE13 = 'QU RMS          '  / Type (heading) of field 13
TUNIT13 = 'JY/BEAM '          / physical units of field 13
TFORM14 = '2E      '          / FORTRAN format of field 14
TTYPE14 = 'WTED QU RMS     '  / Type (heading) of field 14
TUNIT14 = 'JY/BEAM '          / physical units of field 14
YBLC    =            1
YTRC    =          128
ZBLC    =            1
ZTRC    =          128
YINC    =            4
ZINC    =            4
ICLIP   =   0.75000000000000000D+00
PCLIP   =   0.50000000000000000D+00
PIX FIT =            0
END
```

B.8.5 AIPS ST tables

```
XTENSION= 'BINTABLE'          / Extension type
BITPIX  =                   8 / Binary data
NAXIS   =                   2 / Table is a matrix
NAXIS1  =                  56 / Width of table in bytes
NAXIS2  =                   3 / Number of entries in table
PCOUNT  =                   0 / Random parameter count
GCOUNT  =                   1 / Group count
TFIELDS =                   7 / Number of fields in each row
EXTNAME = 'AIPS ST '          / AIPS table file
EXTVER  =                   1 / Version number of table
```

```
TFORM1  = '1D      '           / FORTRAN format of field  1
TTYPE1  = 'RA---SIN        '   / Type (heading) of field  1
TUNIT1  = 'DEGREES '           / Physical units of field  1
TFORM2  = '1D      '           / FORTRAN format of field  2
TTYPE2  = 'DEC--SIN        '   / Type (heading) of field  2
TUNIT2  = 'DEGREES '           / Physical units of field  2
TFORM3  = '1E      '           / FORTRAN format of field  3
TTYPE3  = 'MAJOR AX        '   / Type (heading) of field  3
TUNIT3  = 'DEGREES '           / Physical units of field  3
TFORM4  = '1E      '           / FORTRAN format of field  4
TTYPE4  = 'MINOR AX        '   / Type (heading) of field  4
TUNIT4  = 'DEGREES '           / Physical units of field  4
TFORM5  = '1E      '           / FORTRAN format of field  5
TTYPE5  = 'POSANG          '   / Type (heading) of field  5
TUNIT5  = 'DEGREES '           / Physical units of field  5
TFORM6  = '1E      '           / FORTRAN format of field  6
TTYPE6  = 'STARTYPE        '   / Type (heading) of field  6
TUNIT6  = 'INDEX   '           / Physical units of field  6
TFORM7  = '24A     '           / FORTRAN format of field  7
TTYPE7  = 'LABEL           '   / Type (heading) of field  7
TUNIT7  = 'STRING  '           / Physical units of field  7
END
```

B.8.6 AIPS XG tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  224 / Width of table in bytes
NAXIS2  =               512176 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   11 / Number of fields in each row
EXTNAME = 'AIPS XG '           / AIPS table file
EXTVER  =                    2 / Version number of table
TFORM1  = '2J      '           / FORTRAN format of field  1
TTYPE1  = 'PIXEL           '   / Type (heading) of field  1
TUNIT1  = 'PIXELS  '           / Physical units of field  1
TFORM2  = '1J      '           / FORTRAN format of field  2
TTYPE2  = 'NGAUS           '   / Type (heading) of field  2
TUNIT2  = '        '           / Physical units of field  2
TFORM3  = '1E      '           / FORTRAN format of field  3
TTYPE3  = 'PEAK            '   / Type (heading) of field  3
TUNIT3  = 'JY/BEAM '           / Physical units of field  3
TFORM4  = '2E      '           / FORTRAN format of field  4
TTYPE4  = 'BASELINE        '   / Type (heading) of field  4
TUNIT4  = 'JY/BEAM '           / Physical units of field  4
TFORM5  = '2E      '           / FORTRAN format of field  5
TTYPE5  = 'ERR BASE        '   / Type (heading) of field  5
TUNIT5  = 'PIXELS  '           / Physical units of field  5
TFORM6  = '8E      '           / FORTRAN format of field  6
TTYPE6  = 'AMPLITUD        '   / Type (heading) of field  6
TUNIT6  = 'JY/BEAM '           / Physical units of field  6
TFORM7  = '8E      '           / FORTRAN format of field  7
TTYPE7  = 'CENTER          '   / Type (heading) of field  7
TUNIT7  = 'PIXELS  '           / Physical units of field  7
TFORM8  = '8E      '           / FORTRAN format of field  8
```

```
TTYPE8  = 'WIDTH           '   / Type (heading) of field  8
TUNIT8  = 'PIXELS  '           / Physical units of field  8
TFORM9  = '8E      '           / FORTRAN format of field  9
TTYPE9  = 'ERR AMPL        '   / Type (heading) of field  9
TUNIT9  = 'JY/BEAM '           / Physical units of field  9
TFORM10 = '8E      '           / FORTRAN format of field 10
TTYPE10 = 'ERR CENT        '   / Type (heading) of field 10
TUNIT10 = 'PIXELS  '           / physical units of field 10
TFORM11 = '8E      '           / FORTRAN format of field 11
TTYPE11 = 'ERR WIDT        '   / Type (heading) of field 11
TUNIT11 = 'PIXELS  '           / physical units of field 11
YBLC    =           1
YTRC    =         952
ZBLC    =           1
ZTRC    =         538
YINC    =           3
ZINC    =           3
VCLIP   =   0.19000000320374966D-02
VOFF    =   0.26000000000000000D+05
PIX FIT =      512176
END
```

B.8.7 AIPS ZE tables

```
XTENSION= 'BINTABLE'           / Extension type
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / Table is a matrix
NAXIS1  =                  184 / Width of table in bytes
NAXIS2  =                65536 / Number of entries in table
PCOUNT  =                    0 / Random parameter count
GCOUNT  =                    1 / Group count
TFIELDS =                   10 / Number of fields in each row
EXTNAME = 'AIPS ZE '           / AIPS table file
EXTVER  =                    7 / Version number of table
TFORM1  = '2J      '           / FORTRAN format of field  1
TTYPE1  = 'PIXEL           '   / Type (heading) of field  1
TUNIT1  = 'PIXELS  '           / Physical units of field  1
TFORM2  = '1E      '           / FORTRAN format of field  2
TTYPE2  = 'I PEAK          '   / Type (heading) of field  2
TUNIT2  = 'JY/BEAM '           / Physical units of field  2
TFORM3  = '1E      '           / FORTRAN format of field  3
TTYPE3  = 'GAIN            '   / Type (heading) of field  3
TUNIT3  = 'JY/BEAM '           / Physical units of field  3
TFORM4  = '1E      '           / FORTRAN format of field  4
TTYPE4  = 'ERR GAIN        '   / Type (heading) of field  4
TUNIT4  = '        '           / Physical units of field  4
TFORM5  = '8E      '           / FORTRAN format of field  5
TTYPE5  = 'B  FIELD        '   / Type (heading) of field  5
TUNIT5  = 'PIXELS  '           / Physical units of field  5
TFORM6  = '8E      '           / FORTRAN format of field  6
TTYPE6  = 'ERR BFLD        '   / Type (heading) of field  6
TUNIT6  = 'PIXELS  '           / Physical units of field  6
TFORM7  = '1J      '           / FORTRAN format of field  7
TTYPE7  = 'NGAUS           '   / Type (heading) of field  7
TUNIT7  = '        '           / Physical units of field  7
TFORM8  = '8E      '           / FORTRAN format of field  8
TTYPE8  = 'AMPLITUD        '   / Type (heading) of field  8
```

```
TUNIT8  = 'JY/BEAM '           / Physical units of field  8
TFORM9  = '8E      '           / FORTRAN format of field  9
TTYPE9  = 'CENTER          '   / Type (heading) of field  9
TUNIT9  = 'PIXELS  '           / Physical units of field  9
TFORM10 = '8E      '           / FORTRAN format of field 10
TTYPE10 = 'WIDTH           '   / Type (heading) of field 10
TUNIT10 = 'PIXELS  '           / physical units of field 10
YBLC    =            1
YTRC    =          256
ZBLC    =            1
ZTRC    =          256
DOGAUS  =            1
PIX FIT =        34439
ICLIP   =   0.50000000000000000D+00
END
```

References

[1] Calabretta, M. R. & Greisen, E. W. 2002, "Representations of Celestial Coordinates in FITS," *Astronomy & Astrophysics*, **395**, 1077.

[2] Calabretta, M. R. & Roukema, B. F. 2007, "Mapping on the HEALpix grid," *Monthly Notices of the Royal Astonomical Society*, **381**, 865.

[3] Cotton, W. D., Tody, D. B., and Pence, W. D. "Binary Table Extension to FITS," *Astronomy & Astrophysics Supplement Series*, **113**, 159-166

[4] Greisen, E. W. 1983, "Non-linear Coordinate Systems in AIPS," *AIPS Memo Series*, **27**, National Radio Astronomy Observatory, Charlottesville, Virginia. Also appeared in 1984, Eds. Albrecht, R., Capaccioli, M., *IAU Working Group on Astronomical Image Processing*, Circular Number 10, Space Telescope Science Institute, Baltimore, Maryland.

[5] Greisen, E. W. 1986, "Additional Non-linear Coordinates," *AIPS Memo Series*, **46**, National Radio Astronomy Observatory, Charlottesville, Virginia.

[6] Greisen, E. W. 2003, "FITS: A Remarkable Achievement in Information Exchange," *Information Handling in Astronomy – Historical Vistas*, Heck, A. ed., Kluwer Academic Publishers, Dordrecht, ISBN 1-4040-1178-4, Astrophysics and Space Science Library, **285**, 71.

[7] Greisen, E. W. 2003b, "AIPS, the VLA and the VLBA," *Information Handling in Astronomy – Historical Vistas*, Heck, A. ed., Kluwer Academic Publishers, Dordrecht, ISBN 1-4040-1178-4, Astrophysics and Space Science Library, **285**, 109.

[8] Greisen, E. W. 2011, "The FITS Interferometry Data Interchange Convention — Revised," *AIPS Memo Series*, **114r**, Socorro, New Mexico, USA

[9] Greisen, E. W. & Calabretta, M. R. 2012, "Representations of World Coordinates in FITS," *Astronomy & Astrophysics*, **395**, 1061-1075

[10] Greisen, E. W., Calabretta, M. R., Valdes, F. G., & Allen, S. L. 2005, "Representations of spectral coordinates in FITS," *Astronomy & Astrophysics*, **446**, 747.

[11] Greisen, E. W. & Harten, R. H. 1981, "An Extension of FITS for Groups of Small Arrays of Data," *Astronomy & Astrophysics Supplement Series*, **44**, 371-374

[12] Grosbøl, P., Harten, R. H., Greisen, E. W., and Wells, D. C. 1988, "Generalized Extensions and Blocking Factors for FITS," *Astronomy & Astrophysics Supplement Series*, **73**, 359-364

[13] Hanisch, R. J., Farris, A., Greisen, E. W., Pence, W. D., Schlesinger, B. M., Teuben, P. J., Thompson, R. W., & Warnock III, A. 2001, "Definition of the Flexible Image Transport System," *Astr. & Astrophys.*, **376**, 359.

[14] Hanisch, R. J. & Wells, D. C. 1988, "World coordinate systems representations within the FITS format," notes from a meeting sponsored by the National Aeronautics and Space Administration, Code EZ, Charlottesville, Virginia. Located at
`http://www.cv.nrao.edu/fits/wcs/wcs88.ps.Z`.

[15]Harten, R. H., Grosbøl, P, Greisen, E. W., & Wells, D. C. 1988, "The FITS tables extension," *Astronomy & Astrophysics Supplement Series*, **73**, 365.

[16]IAU 1982, Commission Resolution B1 & B2, *Transactions of International Astronomical Union*, **XVIIIB**, 45.

[17]IAU 1988, Commission Resolution C1, *Transactions of the International Astronomical Union*, **XXB**, 51.

[18]Kogan, L. 1997, "Position Angle of the VSOP Antenna Feed," *AIPS Memo Series*, **93**, Socorro, New Mexico, USA

[19]Pence, W. D. 1992, "FITSIO and FITS file utility software," *Astronomical Data Analysis Software and Systems - I*, Worrall, D. M. , Biemesderfer, C., Barnes, J. Eds., *Astron. Soc. of the Pacific Conference Series*, **25**, 22.

[20]Pence, W. D. 1999, "CFITSIO, v2.0: a new full-featured data interface," *Astronomical Data Analysis Software and Systems - VIII*, Mehringer, D. M., Plante, R. L., Roberts, D. A. Eds., *Astron. Soc. of the Pacific Conference Series*, **172**, 487.

[21]Pence, W. D., Chiappetti, L., Page, C. G., Shaw, R. A., & Stobie, E. 2010, "Definition of the Flexible Image Transport System, version 3.0," *Astronomy & Astrophysics*, **524**, 1-40

[22]Perley, R. 2010, "On Determining Visibilities from Correlation Products," *EVLA Memo Series*, **145**, Socorro, New Mexico, USA

[23]Ponz, J. D., Thompson, R. W., & Muñoz, J. R. 1994, "The FITS image extension," *Astronomy & Astrophysics Supplement Series*, **105**, 53.

[24]Wells, D. C. 2000, "The FITS experience: lessons learned," *Information Handling in Astronomy*, Heck, A. Ed., Kluwer Academic Publishers, Dordrecht, ISBN 0-7923-6494-5, Astrophysics and Space Science Library, **250**, 65.

[25]Wells, D. C., Greisen, E. W., & Harten, R. H. 1981, "FITS: A Flexible Image Transport System," *Astronomy & Astrophysics Supplement Series*, **44**, 363-370