Astronomy & Astrophysics manuscript no. wcs
(DOI: will be inserted by hand later)

November 13, 2002

Representations of world coordinates in FITS

E. W. Greisen[(1)] and M. R. Calabretta[(2)]

[1] National Radio Astronomy Observatory, P. O. Box O, Socorro, NM USA 87801-0387
[2] Australia Telescope National Facility, P. O. Box 76, Epping, NSW 1710 Australia

Received 24 July 2002 / Accepted 9 September 2002

Abstract. The initial descriptions of the FITS format provided a simplified method for describing the physical coordinate values of the image pixels, but deliberately did not specify any of the detailed conventions required to convey the complexities of actual image coordinates. Building on conventions in wide use within astronomy, this paper proposes general extensions to the original methods for describing the world coordinates of FITS data. In subsequent papers, we apply these general conventions to the methods by which spherical coordinates may be projected onto a two-dimensional plane and to frequency/wavelength/velocity coordinates.

Key words. Methods: data analysis – Techniques: image processing – Astronomical data bases: miscellaneous

1. Introduction

The Flexible Image Transport System, or FITS format, was first described by Wells et al. (1981). This format is characterized by a fixed logical record length of 2880 bytes, and the use of an unlimited number of character-format "header" records with an 80-byte, keyword-equals-value substructure. The header is followed by the header-specified number of binary data records, which are optionally followed by extension records of the specified length, but, at that time, of unspecified format. Since then, a number of authors have suggested various types of extensions (e.g. Greisen & Harten 1981; Grosbøl et al. 1988; Harten et al. 1988; Cotton et al. 1995). Because of its great flexibility, the FITS format has been, and continues to be, very widely used in astronomy. In fact, the FITS tape format was recommended (resolution C1) for use by all observatories by Commission 5 at the 1982 meeting of the IAU at Patras (1983) and the General Assembly of the IAU adopted (resolution R11) the recommendations of its commissions, including the FITS resolution. A committee of the NASA/Science Office of Standards and Technology has codified the current state of FITS into a document which has been accepted as the official definition of the standard (Hanisch et al. 2001).

Wells et al. (1981) anticipated the need to specify the physical, or world, coordinates to be attached to each pixel of an N-dimensional image. By *world coordinates*, we mean coordinates that serve to locate a measurement in some multi-dimensional parameter space. They include, for example, a measurable quantity such as the frequency or wavelength associated with each point in a spectrum or, more abstractly, the longitude and latitude in a conventional spherical coordinate system which define a direction in space. World coordinates may also include enumerations such as "Stokes parameters", which do not form an image axis in the normal sense since interpolation along such axes is not meaningful.

Wells et al. (1981) viewed each axis of the image as having a coordinate type and a reference point for which the pixel coordinate, a coordinate value, and an increment were given. Note that this reference point was not required to occur at integer pixel locations nor even to occur within the image. An undefined "rotation" parameter was also provided for each axis. Since there are, in general, more coordinates to be attached to a pixel than there are "real" axes in the N-dimensional image, the convention of declaring axes with a single pixel was also established in both examples given by Wells et al. The keywords defined were

`CRVAL`n	coordinate value at reference point
`CRPIX`n	array location of the reference point in pixels
`CDELT`n	coordinate increment at reference point
`CTYPE`n	axis type (8 characters)
`CROTA`n	rotation from stated coordinate type

A list of suggested values for `CTYPE`n was provided with few of the details actually required to specify coordinates. The units were specified to be The International System of Units "SI" (meters, kilograms, seconds) with the addition of degrees for angles.

The simplicity of this initial description was deliberate. It was felt that a detailed specification of coordinate types was a lengthy and complicated business, well beyond the scope intended for the initial paper. In addition, the authors felt that a detailed specification would probably be somewhat controversial and thus likely to compromise the possibility of wide-

Send offprint requests to: E. Greisen,
e-mail: `egreisen@nrao.edu`

spread agreement on, and use of, the basic structures of the format. Hindsight also suggests that we were rather naive at the time concerning coordinates and it is fortunate that the detailed specification was postponed until greater experience could be obtained.

The descriptions of coordinates in the initial FITS paper are simply inadequate. They provide no description of the meaning of the world coordinates and suggest a rather incomplete list of coordinate types. The use of a single rotation per axis cannot describe any general rotation of more than two axes.

While participating in the development of the AIPS software package of the National Radio Astronomy Observatory, Greisen (1983, 1986) found it necessary to supply additional details to the coordinate definitions for both spectral and celestial coordinates. Since the latter have been widely used, a NASA-sponsored conference held in January 1988 recommended that they form the basis for a more general coordinate standard (Hanisch & Wells 1988); such a standard is described below.

The present work generalizes the set of world coordinate system (WCS) FITS keywords with a view to describing nonlinear coordinate systems and any parameters that they may require. Alternate keywords which should be supported are described. It also addresses the questions of units, multiple coordinate descriptions, uncertainties in the coordinate values, and various other coordinate related matters. Conventions for attaching coordinate information to tabular data are also described. Paper II (Calabretta & Greisen 2002) and Paper III (Greisen et al. 2003) extend these concepts to the ideal, but non-linear angular and spectral coordinates used in astronomy. Paper IV (Calabretta et al. 2003) then provides methods to describe the distortions inherent in the image coordinate systems of real astronomical data. The complex questions related to time systems and to other kinds of coordinates will be deferred.

2. Basic concepts

2.1. Coordinate definition and computation

In the current proposal, we regard the conversion of pixel coordinates to world coordinates as a multi-step process. This is illustrated conceptually in Fig. 1, which shows only the steps to be discussed here. Later extensions may interpose additional steps as required. For example, Paper II divides the final step into two with the computation of intermediate spherical coordinates that are subsequently converted to celestial coordinates via a spherical rotation. Paper IV interposes optional distortion corrections between the first and/or second steps of Fig. 1. Generally these are intended to account for small residuals that cannot be described by one of the standard world coordinate transformations. These may arise in a variety of ways; naturally (e.g. aberration or atmospheric refraction), via complex instrumental response functions (e.g. data cubes produced by a Fabry-Perot interferometer for which surfaces of constant wavelength are curved), by the intrinsic nature of the system under study (e.g. surface coordinates of the asteroid Eros), or as a result of instrumental peculiarities.

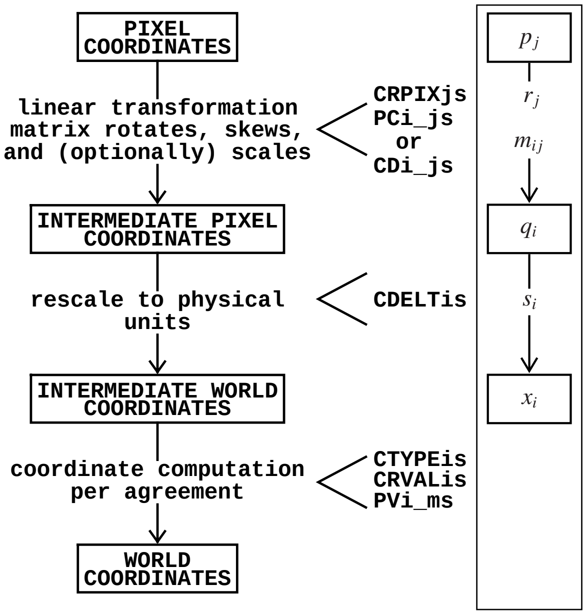

Fig. 1. Conversion of pixel coordinates to world coordinates shown as a multi-step process. In the first step a linear transformation is applied via matrix multiplication of the pixel coordinate vector. This linear transformation may be restricted to the geometrical effects of rotation and skewness with scaling to physical units deferred until the second step (PCi_j plus CDELTi formalism). Alternatively, scaling may be applied via the matrix with the second step omitted (CDi_j formalism). The final step applies a possibly non-linear transformation to produce the final world coordinates. Although generic keywords for this step are defined in this paper, the mathematical details, including the interpretation of the INTERMEDIATE WORLD COORDINATES, are deferred to later papers which may also interpose additional steps in the algorithm chain. For later reference, the mathematical symbols associated with each step are shown in the box at right.

2.1.1. Basic formalism

For all coordinate types, the first step is a linear transformation applied via matrix multiplication of the vector of *pixel coordinate* elements, p_j:

$$q_i = \sum_{j=1}^{N} m_{ij}(p_j - r_j)\,, \tag{1}$$

where r_j are the pixel coordinate elements of the reference point given by the CRPIXj. Henceforth we will consistently use j for pixel axis indexing and i for the world axes.

The m_{ij} matrix is a non-singular square matrix of dimension $N \times N$. In the first instance, N is given by the NAXIS keyword value, but this will be generalized with the introduction of the WCSAXES keyword in Sect. 2.2 so that the dimension of the world coordinates need not match that of the data array.

The elements, q_i, of the resulting *intermediate pixel coordinate* vector are offsets, in dimensionless pixel units, from the reference point along axes coincident with those of the *intermediate world coordinates*. Thus the conversion of q_i to the corresponding intermediate world coordinate element, x_i, is a simple scale:

$$x_i = s_i q_i\,. \tag{2}$$

We defer discussion of the encoding of m_{ij} and s_i as FITS header cards to Sect. 2.1.2.

The third step in the process of computing world coordinates depends on the `CTYPE`i. For simple linear axes, the x_i are interpreted as offsets to be added to the coordinate value at the reference point given by `CRVAL`i. Otherwise, the `CTYPE`i define a function of the x_i, the `CRVAL`i, and, perhaps, other parameters that must be established by convention and agreement. Any `CTYPE`i not covered by convention and agreement shall be taken to be linear.

Non-linear coordinate systems will be signaled by `CTYPE`i in "4-3" form: the first four characters specify the coordinate type, the fifth character is a "-", and the remaining three characters specify an algorithm code for computing the world coordinate value, for example `'ABCD-XYZ'`. We explicitly allow the possibility that the coordinate type may augment the algorithm code, for example `'FREQ-F2W'` and `'VRAD-F2W'` may denote somewhat different algorithms (see Paper III). Coordinate types with names of less than four characters are padded on the right with "-", and algorithm codes with less than three characters are padded on the right with blanks, for example `'RA---UV '`. However, we encourage the use of three-letter algorithm codes.

Particular coordinate types and algorithm codes must be established by convention. Paper II constructs the framework for celestial coordinate systems, and Paper III does so for spectral axes (frequency-wavelength-velocity). `CTYPE`i values that are not in "4-3" form should be interpreted as linear axes. It is possible that there may be old FITS files with a linear axis for which `CTYPE`i is, by chance, in 4-3 form. However, it is very unlikely that it will match a recognized algorithm code (use of three-letter codes will reduce the chances). In such a case the axis should be treated as linear.

2.1.2. Linear transformation matrix

The proposal to replace the `CROTA`i keywords of Wells et al. (1981) with a general linear transformation matrix dates from Hanisch & Wells (1988), although the details of its implementation have undergone considerable evolution. The main point of divergence has been whether the matrix should completely replace or simply augment the `CDELT`i, but there are also important differences relating to the default values of the matrix elements.

In defining a nomenclature which augments the `CDELT`i we have been guided by the following considerations:

- Where possible, standards should grow by generalizing existing usage rather than developing a separate parallel usage. Augmenting the existing `CDELT`i with a separate transformation matrix that defaults to unity makes old headers equivalent to new ones that omit the keywords that define the transformation matrix. In any case, the "once FITS, always FITS" rule means that FITS readers must continue to interpret `CDELT`i, so it makes sense for `CDELT`i to retain its original function.
- The transformation matrix then replaces the poorly defined `CROTA`i with a nomenclature that allows for both skew and fully general rotations. We do not consider this replacement and the consequent deprecation of the `CROTA`i to be inconsistent with the aim of generalizing existing usage since, to our knowledge, the `CROTA`i have had no formal definition other than the "AIPS convention" (Greisen 1983, 1986). Both Wells et al. (1981) and Hanisch et al. (2001) state that "users of this option should provide extensive explanatory comments." Paper II describes the translation of the AIPS interpretation of `CROTA`i to the new formalism.
- A large fraction of WCS representations, perhaps the great majority, will not require the general linear transformation. FITS writers may continue to use `CDELT`i, so FITS-writing software need not be rewritten to conform to the new formalism unless it needs the new features.
- The physical units of a general image may differ by many orders of magnitude, from frequencies of 10^{10} Hz (or more) to angles of 10^{-3} degrees (or less). If the physical units enter into the linear transformation matrix, then the elements of that matrix will have very different magnitudes. These issues pose difficulties both in computing and in understanding, and it may be simpler to defer application of physical units until the multiplication by `CDELT`i.
- These difficulties are compounded when correcting for the distortions present in real instruments. Paper IV will show that some instruments require distortion corrections before, and others after, the linear transformation matrix. Such corrections may need to be expressed in terms directly related to pixel coordinates. If the physical units enter into the linear transformation matrix, then the distortion corrections which come after the matrix would have to compensate for the physical units applied by it, effectively undoing and then redoing a multiplication by `CDELT`i. Furthermore, commensurability problems may arise when recording the maximum distortion correction for a WCS representation that mixes pre-, and post-corrections between axes.
- A widely used formalism that discards `CDELT`i was developed by the Space Telescope Science Institute for the Hubble Space Telescope and was incorporated generally in the IRAF data analysis system. We therefore support this as an alternative method.

In the `PC`i_j formalism, the matrix elements m_{ij} are encoded in

`PC`i_j (floating-valued)

header cards, and s_i as `CDELT`i. The i and j indices are used without leading zeroes, e.g. `PC1_1` and `CDELT1`. The default values for `PC`i_j are 1.0 for $i = j$ and 0.0 otherwise. The `PC`i_j matrix must not be singular; it must have an inverse. Furthermore, all `CDELT`i must be non-zero. In other words, invertibility means that transformations which project from an initial coordinate system of dimensionality `WCSAXES` to a world coordinate system of dimensionality less than `WCSAXES` are forbidden.

In the `CD`i_j formalism Eqs. (1) and (2) are combined as

$$x_i = \sum_{j=1}^{N} (s_i m_{ij})(p_j - r_j) , \tag{3}$$

and the

$$\texttt{CD}i\_j \qquad \text{(floating-valued)}$$

keywords encode the product $s_i m_{ij}$. The i and j indices are used without leading zeroes, e.g. CD1_1. The CDi_j matrix must not be singular; it must have an inverse. CDELTi and CROTAi are allowed to coexist with CDi_j as an aid to old FITS interpreters, but are to be ignored by new readers. The default behavior for CDi_j differs from that for PCi_j; if one or more CDi_j cards are present then all unspecified CDi_j default to zero. If no CDi_j cards are present then the header is assumed to be in PCi_j form whether or not any PCi_j cards are present since this results in an interpretation of CDELTi consistent with Wells et al. (1981).

We specifically prohibit mixing of the PCi_j and CDi_j nomenclatures in any FITS header data unit. With this restriction, translation from the CDi_j formalism to the PCi_j formalism is effected simply in the keyword parsing stage of header interpretation; the CDi_j should be considered equivalent to the PCi_j subject to the considerations for default values noted above and with CDELTi set to unity. Similarly, CDi_j can be calculated from PCi_j and CDELTi following Eq. 3.

2.1.3. Usage comments

The proposal presented in this and the subsequent papers is not simple and provides wide latitude for mistakes in describing the WCS and in writing the FITS headers. The result of an improperly described WCS is simply undefined; it is the job of the FITS writer to produce a correct description. A simple error which could be made in a WCS description, or with other parts of a header, is a repetition of keywords with different values assigned to them. If, for example, BUNIT were repeated with a new value, the data would have unknown units but would be read correctly. In binary tables, a second value for TFORMn would cause the tabular data to be read incorrectly.

This is a very general proposal! The linear transformation matrix allows for skew and fully general rotations. The reader should note that this allows dissimilar axes to be rotated into one another. This is meaningful in imaging; for example, one may wish to re-sample a spectral-line cube from some special viewing angle in the three-space of two celestial coordinates and one frequency coordinate. Such rotations are, however, forbidden into axes whose coordinate values are, by convention, only integral. Thus, if CTYPEi_0 indicates a world coordinate of integral type, then row i_0 of the linear transformation matrix must contain only one non-zero element, and this would normally be 1.0 or at least integral. Additionally, it must be the only non-zero element in the column containing it. The STOKES axis is one such coordinate; see Sect. 5.4.

The linear transformation matrix could also be used to represent images that have been transposed, e.g.

$$\mathrm{PC} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} .$$

This is a legal usage, but likely to confuse the reader. In this example, the FITS user will read in the header that the first element of the world coordinate is CTYPE1, although this corresponds to the second pixel axis. Note that keywords NAXIS1, CRPIX1, PCi_1, and CDi_1, for example, all refer to the first *pixel* axis in the image, while CTYPE1, CRVAL1, PC1_j, CD1_j, and CDELT1 all refer to the first world coordinate ("q_1" and "x_1") element. They must produce a correct result when Eqs. (1) and (2) or Eq. (3) are applied. Thus, x_1 is of type CTYPE1 even if it does not change with p_1 (to use the nomenclature of Eqs. (1), (2), and (3)). Therefore, it is good form to transpose the header parameters along with the image so that the on-diagonal terms in the transformation matrix predominate. If the PC or CD matrix is essentially diagonal, then the human reader of the FITS header will have a better chance of understanding the coordinate representation.

Equations (1) and (2) allow considerable flexibility in the way the linear transformation is partitioned between the PCi_j and CDELTi. In the absence of any formal constraints, the normal expectation would be that the CDELTi be used as scaling parameters as in the past. This is straightforward if PCi_j is orthogonal, i.e. defines a pure rotation or simple reflection, but not if it has an element of skewness. In general, a reasonable approach is to choose CDELTi so that

$$\sum_{j=1}^{N} \texttt{PC}i\_j^{\,2} = 1 \,. \tag{4}$$

for all i. This normalization leaves orthogonal matrices unchanged, and only slightly modifies matrices which are nearly orthogonal. Note that this is not the same as setting the determinant of the PCi_j matrix to unity. Note also that this constraint is optional and may not be the most physically meaningful selection of the PCi_j. For example, the conversion from the old CROTAi nomenclature to the new PCi_j form described in Paper II does not satisfy this constraint unless the CDELTi are equal.

2.1.4. Additional points

Note that integer pixel numbers refer to the center of the pixel in each axis, so that, for example, the first pixel runs from pixel number 0.5 to pixel number 1.5 on every axis. Note also that the reference point location need not be integer nor need it even occur within the image. The original FITS paper (Wells et al. 1981) defined the pixel numbers to be counted from 1 to NAXISj (≥ 1) on each axis in a Fortran-like order as presented in the FITS image.[1]

[1] This convention differs from the usual practice in computer graphics where the pixels are counted from zero with pixel centers as half integers (e.g. Adobe Systems, Inc. 1999). The convention proposed here has been used extensively in FITS since the format was invented and no argument has been advanced sufficiently compelling to invalidate the many thousands of files written with that convention. Furthermore, we regard our image samples as "voxels" in real physical space rather than pixels in two-dimensional display space. As such, they may be viewed from any angle via transposition and rotation. The only point within the individual voxel that remains invariant under those operations is its center and we argue, therefore, that it is the center of the voxel which we should count.

A WCS representation should be invertible in the sense that a pixel coordinate, when transformed to a world coordinate, must be uniquely recoverable from that world coordinate. Note that this does not require that each pixel coordinate in an image have a valid world coordinate; as an example, pixel coordinates in the corner of a Hammer-Aitoff projection of the full sky fall outside the map boundary. Nor need each valid world coordinate correspond to a pixel coordinate; for example, the divergent poles of the Mercator projection are inaccessible. In practical terms, it means that two or more different pixel coordinates should not map to the same world coordinate, as exemplified by a cylindrical projection in which the longitude spans more than 360°. Such coordinate systems, while easy to construct, may be very difficult to interpret properly in all respects, including that of drawing a coordinate grid. Thus, while they are not explicitly prohibited, it may be expected that general WCS interpreting software may not handle them properly.

An additional convention is needed where non-linear axes must be grouped, for example, the two axes which form a map plane. In general, all axes in the group must have identical algorithm codes and a scheme must be established by convention for associating members of the group and, if necessary, their order. For example, Paper II introduces the '*x*LON/*x*LAT' and '*yz*LN/*yz*LT' conventions for associating longitude/latitude coordinate pairs. This should serve as a model for other cases. Note that grouping is not required for linear axes which are always separable (in the mathematical sense) by means of a rotation or skew applied via the linear transformation matrix.

Some non-linear algorithms require parameter values, for example, conic projections require the latitudes of the two standard parallels. Where necessary, numeric parameter values will be specified via

PV*i*_*m* (floating-valued)

keywords, where i is the intermediate world coordinate axis number and m is the parameter number. Leading zeros are not allowed and m may have only those values defined for the particular non-linear algorithm in the range 0 through 99 only. There may also be auxiliary keywords which are required to define, for example, the frames of reference used for celestial and velocity coordinates.

A few non-linear algorithms may also require character-valued parameters, for example, table lookups require the names of the table extension and the columns to be used. Where necessary, character-valued parameters will be specified via

PS*i*_*m* (character-valued)

keywords, where i is the intermediate world coordinate axis number and m is the parameter number. Leading zeros are not allowed and m may have only those values defined for the particular non-linear algorithm in the range 0 through 99 only.

The keywords proposed above and throughout the main body of this manuscript apply to the relatively simple images stored in the main FITS image data and in IMAGE extensions Ponz et al. 1994). When coordinates are used to describe image fragments in BINTABLE extension tables (Cotton et al. 1995), additional nomenclature conventions are required. These are described in Sect. 3.

2.2. Coordinate dimensionality

The number of world coordinate elements associated with a datum can exceed the number of pixel coordinate elements which locate it in the image data array. For example, long-slit optical spectra are naturally two-dimensional; normally the slit is aligned with one (spatial) pixel axis and the dispersion coincides with the other (spectral) pixel axis. While the spectral representation is straightforward – one spectral pixel coordinate transforms to just one spectral world coordinate (frequency, wavelength or velocity) – the spatial representation would appear to be problematic. Since the slit can be oriented at any angle on the sky, the single pixel coordinate which locates a datum along the length of the slit must transform to two spatial (angular) coordinates, typically a right ascension and a declination, neither of which need be constant.

In fact, this problem was solved very early in the history of FITS. Wells et al. (1981) illustrate headers containing degenerate axes, i.e. axes having NAXISj = 1, and this, combined with the meaning assigned to CROTAi by Greisen (1983), provides a fully functional solution. While not previously documented outside the AIPS project, this solution is well known and has been used extensively. Basically the idea is simply to increment the number of pixel coordinate elements as required by introducing degenerate axes.

For the long slit example, we set NAXIS = 3 and NAXIS3 = 1. Supposing without loss of generality that CTYPE1 is the spectral axis, we represent CTYPE2 as right ascension and CTYPE3 as declination. CROTAi in the original formulation is here replaced by PC*i*_*j*, so that pixel coordinates along the length of the slit, $(p_2, p_3 = 1)$, transform to intermediate world coordinates (x, y). Thus the slit's locus in the xy-plane is a straight line whose orientation can be controlled via the PC*i*_*j* matrix. Details of the transformation of (x, y) to celestial spherical coordinates are properly the subject of Paper II. However, given that the xy-plane is to be interpreted as the map plane of a spherical projection, it should be clear that rotating the slit in the xy-plane via the PC*i*_*j* matrix corresponds to rotating it on the sky. Paper II discusses this long slit example in detail.

There is concern that requiring the use of degenerate pixel axes would have severe repercussions for a significant fraction of existing software programs which were not written to deal with such usage. For example, some software intended to handle two-dimensional images would reject a FITS header with NAXIS = 3, even if the third axis is degenerate. At the same time, degenerate axes are a widespread and natural representation for images in a multi-dimensional space. Furthermore, as shown in Fig. 1 of Wells et al. (1981), explicit specification of degenerate axes allows them to appear in any order. Such usage may facilitate image building and sub-imaging operations.

To provide a solution for this world-coordinate dimensionality problem that does not require the use of degenerate axes, we reserve the keyword

WCSAXES (integer-valued)

to specify the highest value of the index of any WCS keyword in the header (i.e. CRPIXj, PC*i*_*j* or CD*i*_*j*, CDELTi, CTYPEi, CRVALi, or CUNITi). The default value is the larger of NAXIS and the largest index of these keywords found in the FITS header. This keyword, if present, must precede all WCS keywords (other than the NAXISj). The use of this keyword also solves the problem posed by alternate axis descriptions (Sect. 2.5) which may have an intrinsically different coordinate dimensionality. In the slit example, an alternate description of the x, y coordinates on the detector has no use for a third axis. It is a good idea to provide a coordinate description, even if it is only a "pixel" axis, for all array axes having more than one pixel.

There is debate within the community as to whether the official definition of FITS (Hanisch et al. 2001) prohibits the occurrence of WCS-related keywords with indices greater than the value of NAXIS. We make no claims one way or the other, but rather assert that in order to accommodate WCS specifications whose dimensionality exceeds NAXIS without the use of degenerate coordinate axes, such use must be allowed. Consistent with Hanisch et al. (2001), however, no NAXISj keywords may exist for j > NAXIS. Thus, calculations related to determining the total length of the data array, which relies upon a product of the NAXISj values, are unaffected. Accordingly, all axes with axis number greater than NAXIS must be one pixel in length implicitly rather than explicitly.

2.3. Keyword value units

The original FITS paper (Wells et al. 1981) assumed that the units along each axis could be implied simply by the contents of the CTYPEi keyword. This has not turned out to be true in general. Therefore, we propose adding a new indexed, keyword

CUNITi (character-valued)

with which the units of CRVALi and CDELTi may be specified. Restrictions on the nature and range of units, if any, will be determined by the agreements applying to the specific axes. If they are not so limited, units should conform with the recommendations of the IAU Style Manual (McNally 1988). Particular conventions for CUNITi values are discussed in Sect. 4. Case will be significant in the values assigned to CUNITi since, for example, it is necessary to represent both milli and Mega prefixes for units such as mJy and MJy. The values assigned to CUNITi cannot exceed 68 characters, but may well be longer than the 8 characters which has been a traditional – but optional – limit for character-valued but non-mandatory keywords.

2.4. Keyword value defaults

The original FITS paper also assumed that the coordinate keywords, if present, would all be present and, therefore did not define defaults for the standard keywords. We therefore define the defaults to be

WCSAXES	NAXIS or largest i or j
CRVALi	0.0
CRPIXj	0.0
CDELTi	1.0
CTYPEi	' ' (i.e. a linear undefined axis)
CUNITi	' ' (i.e. undefined)
PCi_j	1.0 when $i = j$
PCi_j	0.0 when $i \neq j$
CDi_j	0.0 but see Sect. 2.1.2

These defaults provide the minimal amount of information consistent with a real axis that is not fully described but will not cause zero divides. They assert that the pixel coordinate value changes as the pixel number changes and that, by default, coordinate values on the pixel axis depend upon that axis and only that axis. The reference pixel is by default just off the data array which satisfies the needs of some software systems and has the felicitous result that the coordinate value of each pixel is its pixel number if all of the keywords take their default values. With these defaults, a program may fill its coordinate arrays with usable, if uninteresting, values before reading the FITS header, rather than constructing some scheme that changes depending on the *absence* of keywords in the FITS header. Because default values were not defined from the beginning and appear to be a source of confusion, we recommend that FITS writers should endeavor always to write complete WCS specifications and never to depend upon defaults.

2.5. Alternate axis descriptions

In some cases, an axis of an image may be described as having more than one coordinate type. An example of this would be the frequency, velocity, and wavelength along a spectral axis (only one of which, of course, could be linear). One can also describe the position on a CCD camera chip (or photographic plate) in meters as well as in degrees on the sky. To allow up to 26 additional descriptions of each axis, we propose the addition of the following optional, but now reserved, keywords defined in Table 1, where j and i are pixel and intermediate world coordinate axis numbers, respectively, and a is a character A through Z specifying the coordinate version. The axis numbers are restricted by this convention to the range 1–99 and the coordinate parameter m is restricted to the range 0–99, both with no leading zeros. Note that the primary version of the coordinate description is that specified with a as the blank character. If an alternate coordinate description is specified, all coordinate keywords for that version must be given even if they do not differ from those of the primary version. Rules for the default values of alternate coordinate descriptions are the same as those for the primary description. The alternate coordinate descriptions are computed in the same fashion as the primary coordinates. The type of coordinate depends on the value of CTYPEia and may be linear in one of the alternate descriptions and non-linear in another.

Alternate axis descriptions are optional, but may only be specified if a primary axis description is specified. The alter-

Table 1. Keywords with alternate axis descriptor codes.

WCSAXESa	number of axes in WCS description (integer)
CRVALia	coordinate value at reference point (real floating)
CRPIXja	pixel coordinate of the reference point (real floating)
PCi_ja	linear transformation matrix (real floating)
CDELTia	coordinate increment (real floating)
CDi_ja	linear transformation matrix (with scale) (real floating)
CTYPEa	axis type (8 characters)
CUNITia	units of CRVALia and CDELTia (character)
PVi_ma	coordinate parameter m (real floating)
PSi_ma	coordinate parameter m (character)

nate version codes are selected by the FITS writer; there is no requirement that the codes be used in alphabetic sequence and no requirement that one coordinate version differ in its parameter values from another.

An optional keyword

WCSNAMEa (character-valued)

is also defined to name, and otherwise document, the various versions of world coordinate descriptions. This keyword can be used to give the user simple names by which to request the various versions of the coordinates. It may also be used, for example, to distinguish coordinates used during data acquisition from those determined later by astrometrically rigorous reductions. It might also be used to specify which are data pixels and which are calibration pixels in a CCD image.

2.6. Uncertainties in the coordinates

The coordinates of a pixel may not always be known exactly. Instead, they are often subject to both random, statistical errors and various systematic errors. The former are not particularly correlated between pixels, whereas the latter may have a high degree of correlation across the whole data set. For example, single-dish radio images may be accurate on a pixel-to-pixel basis to a fraction of an arcsec, but have a 5–10 arcsec uncertainty in the reference point value. Two optional keywords are hereby reserved to specify these uncertainties in coordinates. They are

CRDERia	random error in coordinate (real floating)
CSYERia	systematic error in coordinate (real floating)

where both are given in units of CUNITia and have default value zero. They are understood to give a representative average value of the error over the range of the coordinate in the data file. The total error in the coordinate would be given by summing the two errors in quadrature.

The errors in actual coordinates may be very much more complex than this simple representation. In the most general case, one might require, at each pixel, a covariance matrix to describe the dependence of the uncertainty in one coordinate on the uncertainties in the others. Furthermore, the errors in one coordinate description may, or may not, be completely predictable from those of an alternate description. Such usages, while perhaps important under some circumstances, are well beyond the needs of most users and the scope of this manuscript.

3. Alternate FITS image representations: pixel list and vector column elements[2]

In addition to the image format discussed in the previous sections of this paper (i.e. an N-dimensional array in a FITS primary array or FITS IMAGE extension), there are two other FITS image representations that are used commonly by the astronomical community in binary tables extensions (Cotton et al. 1995) in the forms of:

1. a multi-dimensional vector in a single element of a FITS binary table,
2. a tabulated list of pixel coordinates in a FITS ASCII or binary table, and
3. a combination of the two forms in a FITS binary table.

The purpose of this section is to define a naming convention for the coordinate system keywords to be used with these alternate image formats. Keywords specific to celestial coordinates will be treated in Paper II and an example will be given. Keywords specific to spectral coordinates will be treated in a section of Paper III. This general convention has been used for some time and is therefore considered part of the full world coordinates convention.

The NOST (Hanisch et al. 2001) standard provides that the interpretation of raw field values found in any column n of a FITS table (either ASCII or binary) may be transformed into true physical values by the presence of the keywords TZEROn and TSCALn for that column. The tabular WCS keywords defined in this section (and in the corresponding tabular keyword sections of subsequent WCS papers) operate on the true physical values, not on the raw field values. Therefore any transformation specified by TZEROn and TSCALn is to be applied before these tabular WCS computations.

3.1. Multi-dimensional vector in a binary table

A vector column in a binary table (BINTABLE) extension can be used to store a multi-dimensional image in each element (i.e. each row) of the column. In the simple case in which all the images have the same fixed size, the TDIMn keyword can be used to specify the dimensions. In the more general case,

[2] Contributed by William Pence, Arnold Rots, and Lorella Angelini of the NASA Goddard Space Flight Center, Greenbelt, MD 20771.

Table 2. Coordinate keywords for use in tables: the data type of the table keyword matches that of the primary array keyword. See Sect. 3.3 for the definitions of the italicized metasyntactic variables used below.

Keyword Description	Primary Array	BINTABLE vector primary	BINTABLE vector alternate	Pixel List primary	Pixel List alternate
Coordinate dimensionality	WCSAXES*a*	WCAX*na*		–	
Axis Type	CTYPE*ia*	*i*CTYP*n*	*i*CTY*na*	TCTYP*n*	TCTY*na*
Axis Units	CUNIT*ia*	*i*CUNI*n*	*i*CUN*na*	TCUNI*n*	TCUN*na*
Reference value	CRVAL*ia*	*i*CRVL*n*	*i*CRV*na*	TCRVL*n*	TCRV*na*
Coordinate increment	CDELT*ia*	*i*CDLT*n*	*i*CDE*na*	TCDLT*n*	TCDE*na*
Reference point	CRPIX*ja*	*j*CRPX*n*	*j*CRP*na*	TCRPX*n*	TCRP*na*
Transformation matrix	PC*i_ja*	*ij*PC*na*		TP*n_ka*	
Transformation matrix	CD*i_ja*	*ij*CD*na*		TC*n_ka*	
Coordinate parameter	PV*i_ma*	*i*V*n_ma*		TV*n_ma*	
Coordinate parameter array	–	*i*V*n*_X*a*		–	
Coordinate parameter	PS*i_ma*	*i*S*n_ma*		TS*n_ma*	
Coordinate name	WCSNAME*a*	WCSN*na*		TWCS*na*	
Random error	CRDER*ia*	*i*CRD*na*		TCRD*na*	
Systematic error	CSYER*ia*	*i*CSY*na*		TCSY*na*	
WCS cross-ref. target	–	WCST*na*		–	
WCS cross reference	–	WCSX*na*		–	
† Coordinate rotation	CROTA*i*	*i*CROT*n*		TCROT*n*	

† CROTA*i* form is deprecated. It may be used only when PC*i_j*, PV*i_m*, and PS*i_m* are not used and when *a* is blank.

a variable length vector may be used to store different-sized images within the same column.

Because two or more columns in a binary table can contain images, the naming convention for these coordinate system keywords must encode the column number containing the image to which the keyword applies as well as the axis number within the image. The naming convention described here uses the keyword prefix to specify the axis number and the keyword suffix to specify the column number containing the image (e.g. the 2CRVL15 keyword applies to the second axis of the image in Col. 15 of the table).

3.2. Tabulated list of pixels

An image may also be represented as a list of $p_1, p_2, \ldots$ pixel coordinates in a binary or ASCII table extension. This representation is frequently used in high-energy astrophysics as a way of recording the position and other properties of individually detected photons. This image format requires a minimum of n table columns which give the $p_1, p_2, \ldots, p_n$ (axes 1 through n) pixel coordinate of the corresponding event in the virtual n-D image; any number of other columns may be included in the table to store other parameters associated with each event such as arrival time or photon energy. This virtual image may be converted into a real image by computing the n-dimensional histogram of the number of listed events that occur in each pixel of the image (i.e. the intensity value assigned to each pixel $(p_1, p_2, \ldots, p_n)$ of the image is equal to the number of rows in the table which have axis 1 coordinate = p_1, axis 2 coordinate = p_2), etc.

A variation on this pixel list format may be used to specify explicitly the intensity value of each image pixel. This case requires at least $n + 1$ table columns which specify the axis 1 coordinate, the axis 2 coordinate, etc. plus the value of the pixel at that coordinate. In this representation each pixel coordinate would only be listed at most once in the table; pixels with a value = 0 may be omitted entirely from the table to conserve space.

Each axis of the image in this representation translates into a separate column of the table, so the suffix of the coordinate system keywords all refer to a column number rather than an axis number (e.g. the TCRP12 keyword applies to the coordinates listed in the 12th column of the table). This form of WCS keyword may only be used with columns containing scalar values; the BINTABLE form must be used with columns containing more than one value per table cell.

The presence of data from each column within a particular row of a table implies an association of those data with each other. However, no formal method has previously been defined for identifying and associating the columns which contain image pixel coordinates, although informal conventions using new keywords have been used. Past practice has been to use distinctive column names (e.g. TTYPE*n* keywords with values of 'DETX' and 'DETY', or 'X' and 'Y') which a human interpreter may use to form associations. However, this is not generally suitable for interpretation by software.

The keywords defined for pixel lists in Table 2 partially remedy this by identifying the pixel coordinate columns. For example, the presence of TCTY*na* in the header of a binary table identifies column *n* as containing a pixel coordinate rather than, say, a pixel value. In so doing it also identifies the binary table as a pixel list. Moreover, where a pixel list contains multiple coordinate representations, the presence of a complete set of TP*n_ka* keywords would also provide a method of associating the coordinate axes of each representation.

Thus, pending a formal solution of this problem, it is suggested that a complete set of TP*n*_*ka* (or TC*n*_*ka*) keywords be included in the pixel list header to define an association of coordinate axes. It should be noted that, while such an association is unordered, this is not a concern for the computation of world coordinates.

3.3. Keyword naming convention

Table 2 lists the corresponding set of coordinate system keywords for use with each type of FITS image representation. The data type of the table keyword matches that of the corresponding primary image keyword. The allowed values for these keywords are identical for all three types of images as defined in the main body of this paper. The old and now deprecated keyword CROTA*i* has been used with tables and is included since readers will need to understand this keyword even if writers should no longer write it. See Paper II for a discussion of this point. To support current usage, the keywords are given in their current form to be used for the primary coordinate representation (*a* is blank) and a new form to support the new capability to specify alternate coordinates for the same axis (*a* is A through Z). For new keywords, the two forms are identical and are shown in a single column midway between the columns for old primary and new alternate WCS keywords. The following notes apply to the naming conventions used in Table 2:

- The *j*, *i* prefix and suffix characters are integers referring to a pixel and intermediate world coordinate axis number, respectively, of the array. When used as a keyword suffix the image dimension may range from 1 to 99 with no leading 0, but when used as a prefix the integer is limited to a single digit to conform to the 8-character keyword name limit so the image may only contain up to 9 dimensions.
- *a* is a 1-character coordinate version code and may be blank (primary) or any single uppercase character from A through Z.
- *n* and *k* are integer table column number without any leading zeros (1–999).
- *m* is an integer between 0 and 99 with no leading zero giving the coordinate parameter number. *m* cannot exceed 9 when *n* exceeds 99, but see the following Section.
- The guidelines given Sect. 2.3 must be applied to the the value of the CUNIT*ia* keyword and its derivatives. In particular the value is restricted to `'deg'` when referring to celestial coordinates; see Paper II.

3.4. Multiple images and the "Greenbank Convention"

In the case of the binary table vector representation, all the images contained in a given column of the table may not necessarily have the same coordinate transformation values. For example, the pixel location of the reference point may be different for each image/row in the table, in which case a single 1CRP*n* keyword in the header is not sufficient to record the individual value required for each image. In such cases, the keyword must be replaced by a column with the same name (i.e. TTYPE*m* = '1CRP*n*') which can then be used to store the pixel location of the reference point appropriate for each row of the table. This convention for expanding a keyword into a table column (or conversely, collapsing a column of identical values into a single header keyword) is commonly known as part of the "Greenbank Convention" for FITS keywords and is illustrated in the example header shown in Paper II (Table 9).

There are several restrictions which may be too limiting for the parameters of certain types of coordinates and, in particular, for the distortion parameters to be introduced in Paper IV. The limitation to 8 characters limits the number of columns to 999, the number of axes to 9, and the number of parameters to as few as 10 (numbered 0 through 9, for column numbers exceeding 99). To avoid this difficulty, we introduce the concept of a parameter array as a single column of a table. All the parameters of a coordinate are given up to the maximum dimension of the column (given by keyword TFORM*n*) with no omitted parameters. Such parameter arrays are signaled by replacing the _*m* in the table column name with _X.

The Greenbank and parameter-array conventions are not needed with pixel lists since they are used to represent a single image.

3.5. Coordinate system cross-references

While a coordinate representation may be shared amongst image arrays within the same *column* of a binary table, it may also happen that several image arrays within the same *row* of a binary table must share the same coordinate representation. For example, each row of a table might store a raw optical spectrum, the corresponding sky background spectrum, a flux-calibrated spectrum derived from these, and a spectrum of the error in each channel. It would not be appropriate to coerce these into a 2-dimensional data array with a heterogeneous second axis, and in any case this would complicate the addition or removal of spectra, say, as the result of data reduction. It also may not be satisfactory simply to repeate the coordinate description for each spectrum. For example, it would be preferable to apply a wavelength calibration to one shared representation rather than several identical copies.

This situation is handled by introducing coordinate system cross-references. These apply only to binary tables containing multiple image array columns, they are not relevant to primary image arrays or to pixel lists which represent only a single data set.

Coordinate system cross-references allow an image array in one column to reference the coordinate system defined for an image array in another column. The cross-reference is specified by the keyword pair

WCST*na*   (character-valued)

for the referred-to (target) coordinate system, and

WCSX*na*   (character-valued)

for the referring (cross-referencing) system, and these must have identical, case-sensitive, keyword values. WCSX*na* must not be combined with any Table 2 coordinate keywords that use the same alternate descriptor. With regard to the Greenbank

Table 3. Characters & strings allowed to denote mathematical operations.

String	Meaning
`str1 str2`	Multiplication
`str1*str2`	Multiplication
`str1.str2`	Multiplication
`str1/str2`	Division
`str1**expr`	Raised to the power `expr`
`str1^expr`	Raised to the power `expr`
`str1expr`	Raised to the power `expr`
`log(str1)`	Common Logarithm (to base 10)
`ln(str1)`	Natural Logarithm
`exp(str1)`	Exponential ($e^{\texttt{str1}}$)
`sqrt(str1)`	Square root

Table 4. Prefixes for multiples & submultiples.

Submult	Prefix	Char	Mult	Prefix	Char
10^{-1}	deci	`d`	10	deca	`da`
10^{-2}	centi	`c`	10^{2}	hecto	`h`
10^{-3}	milli	`m`	10^{3}	kilo	`k`
10^{-6}	micro	`u`	10^{6}	mega	`M`
10^{-9}	nano	`n`	10^{9}	giga	`G`
10^{-12}	pico	`p`	10^{12}	tera	`T`
10^{-15}	femto	`f`	10^{15}	peta	`P`
10^{-18}	atto	`a`	10^{18}	exa	`E`
10^{-21}	zepto	`z`	10^{21}	zetta	`Z`
10^{-24}	yocto	`y`	10^{24}	yotta	`Y`

convention (Sect. 3.4), when `WCST`*na* and/or `WCSX`*na* are columns of the table, the scope of the keyword(s) is limited to one row of the table. Thus the same value of `WCST`*na* and `WCSX`*na* may be reused in different rows.

On encountering `WCSX`*na*, FITS header-parsing software must resolve the reference by searching for `WCST`*na* with the same value, extracting the column number and alternate descriptor suffix encoded in the keyword itself, and then searching for and loading the associated coordinate keywords.

To continue the example above, suppose that Col. 12 contains a raw optical spectrum for which the coordinate system is fully specified, and that `WCST12B` has been set to `'XREF1'`. Then a sky background spectrum in Col. 13 might reference this coordinate system by setting `WCSX13A = 'XREF1'`. In this case, Col. 13 must not contain any of the Table 2 keywords for alternate descriptor a = `A`, although it might contain keywords for some other value of a. The example illustrates that the alternate descriptor suffixes for `WCST`*na* and `WCSX`*na* need not match; the association is via the keyword value alone.

Table 5. IAU-recommended basic units.

Quantity	Unit String	Meaning	Notes
SI base & supplementary units			
length	`m`	meter	
mass	`kg`	kilogram	`g` gram okay
time	`s`	second	
plane angle	`rad`	radian	
solid angle	`sr`	steradian	
temperature	`K`	kelvin	
electric current	`A`	ampere	
amount of substance	`mol`	mole	
luminous intensity	`cd`	candela	
IAU-recognized derived units			
frequency	`Hz`	hertz	s^{-1}
energy	`J`	joule	N m
power	`W`	watt	J s^{-1}
electric potential	`V`	volt	J C^{-1}
force	`N`	newton	kg m s^{-2}
pressure, stress	`Pa`	pascal	N m^{-2}
electric charge	`C`	coulomb	A s
electric resistance	`Ohm`	ohm	V A^{-1}
electric conductance	`S`	siemens	A V^{-1}
electric capacitance	`F`	farad	C V^{-1}
magnetic flux	`Wb`	weber	V s
magnetic flux density	`T`	tesla	Wb m^{-2}
inductance	`H`	henry	Wb A^{-1}
luminous flux	`lm`	lumen	cd sr
illuminance	`lx`	lux	lm m^{-2}

4. Specification of units

Unless agreed otherwise, units should conform with the recommendations of the IAU Style Manual (McNally 1988). Unfortunately, this manual defines units as they would appear in a published document rather than as they must appear in plain character form. George & Angelini (1995) and Ochsenbein et al. (1996) have prepared detailed documents on this subject upon which the following remarks are based. In particular, the following tables are taken from George & Angelini's manuscript (with some changes and additions). Readers should consult these references for examples and expanded discussion. We allow the possibility that the interpretation of `CUNIT`*ia* may depend on conventions established for the `CTYPE`*ia* associated with them. For example, units specified by "s" might mean SI seconds or seconds of sidereal time.

The basic units string, called `str1` and `str2` in Table 3, is composed of a unit string taken from Col. 2 of the IAU-

Table 6. Additional allowed units.

Quantity		Unit String	Meaning	Notes
plane angle		`deg`	degree of arc	$\pi/180$ rad
		`arcmin`	minute of arc	$1/60$ deg
		`arcsec`	second of arc	$1/3600$ deg
		`mas`	milli-second of arc	$1/3600000$ deg
time		`min`	minute	
		`h`	hour	
		`d`	day	86400 s
	†	`a`	year (Julian)	31557600 s (365.25 d), peta a (`Pa`) forbidden
	†	`yr`	year (Julian)	`a` is IAU-style
energy*	†	`eV`	electron volt	$1.6021765 \times 10^{-19}$ J
	‡	`erg`	erg	10^{-7} J
		`Ry`	rydberg	$\frac{1}{2}\left(\frac{2\pi e^2}{hc}\right)^2 m_e c^2 = 13.605692$ eV
mass*		`solMass`	solar mass	1.9891×10^{30} kg
		`u`	unified atomic mass unit	$1.6605387 \times 10^{-27}$ kg
luminosity		`solLum`	Solar luminosity	3.8268×10^{26} W
length	‡	`Angstrom`	angstrom	10^{-10} m
		`solRad`	Solar radius	6.9599×10^{8} m
		`AU`	astronomical unit	1.49598×10^{11} m
		`lyr`	light year	9.460730×10^{15} m
	†	`pc`	parsec	3.0857×10^{16} m
events		`count`	count	
		`ct`	count	
		`photon`	photon	
		`ph`	photon	
flux density	†	`Jy`	jansky	10^{-26} W m^{-2} Hz^{-1}
	†	`mag`	(stellar) magnitude	
	†	`R`	rayleigh	$10^{10}/(4\pi)$ photons m^{-2} s^{-1} sr^{-1}
magnetic field	†‡	`G`	gauss	10^{-4} T
area		`pixel`	(image/detector) pixel	
		`pix`	(image/detector) pixel	
	†‡	`barn`	barn	10^{-28} m^2
			Miscellaneous “units”	
		`D`	debye	$\frac{1}{3} \times 10^{-29}$ C.m
		`Sun`	relative to Sun	e.g. abundances
		`chan`	(detector) channel	
		`bin`	numerous applications	(including the 1-d analogue of pixel)
		`voxel`	3-d analogue of pixel	
	†	`bit`	binary information unit	
	†	`byte`	(computer) byte	8 bit
		`adu`	Analog-to-digital converter	
		`beam`	beam area of observation	as in Jy/beam

† - addition of prefixes for decimal multiples and submultiples are allowed.
‡ - deprecated in IAU Style Manual (McNally 1988) but still in use.
∗ - conversion factors from CODATA Internationally recommended values of the fundamental physical constants 1998(`http://physics.nist.gov/cuu/Constants/`)

recognized units in Table 5 or the extended astronomical units in Table 6. All units from the former and selected units from the latter may be preceded, with no intervening blanks by a single character (two for deca) taken from Table 4 and representing scale factors mostly in steps of 10^3. Compound prefixes (e.g., `ZYeV` for 10^{45} eV) are prohibited. A compound string may then be created from these simple strings by one of the notations in Table 3. A unit raised to a power is indicated by the unit string followed, with no intervening blanks, by the optional symbols `**` or `^` followed by the power given as a numeric expression, called `expr` in Table 3. The power may be a simple integer, with or without sign, optionally surrounded by parentheses. It may also be a decimal number (e.g., 1.5, .5) or a ratio of two integers (e.g. 7/9), with or without sign, which are always surrounded by parentheses. Thus meters squared is indicated by `m**(2)`, `m**+2`, `m+2`, `m2`, `m^2`, `m^(+2)`, etc. and per meter cubed is indicated by `m**-3`, `m-3`, `m^(-3)`, `/m3`, and so forth. Meters to the three halves may be indicated by `m(1.5)`, `m^(1.5)`, `m**(1.5)`, `m(3/2)`, `m**(3/2)`, and `m^(3/2)`, but *not* by `m^3/2` or `m1.5`.

Note that functions such as `log` actually require dimensionless arguments, so, by `log(Hz)`, for example, we actually mean `log(`x`/1Hz)`. The final string to be given as the value of `CUNIT`*ia* is the compound string, or a compound of compounds, preceded by an optional numeric multiplier of the form `10**`k, `10^`k, or `10`$\pm k$ where k is an integer, optionally surrounded by parentheses with the sign character required in the third form in the absence of parentheses. FITS writers are encouraged to use the numeric multiplier only when the available standard scale factors of Table 4 will not suffice. Parentheses are used for symbol grouping and are strongly recommended whenever the order of operations might be subject to misinterpretation. A blank character implies multiplication which can also be conveyed explicitly with an asterisk or a period. Therefore, although blanks are allowed as symbol separators, their use is discouraged. Two examples are `'10**(46)erg/s'` and `'sqrt(erg/pixel/s/GHz)'`. Note that case is significant throughout. The IAU style manual forbids the use of more than one solidus (`/`) character in a units string. In the present conventions, normal mathematical precedence rules are assumed to apply, and we, therefore, allow more than one solidus. However, authors might wish to consider, for example, `'sqrt(erg/(pixel.s.GHz))'` instead of the form given previously.

5. Additional matters

5.1. Image display conventions

It is very helpful to adopt a convention for the display of images transferred via the FITS format. Many of the current image processing systems have converged upon such a convention. Therefore, we recommend that FITS writers order the pixels so that the first pixel in the FITS file (for each image plane) be the one that would be displayed in the lower-left corner (with the first axis increasing to the right and the second axis increasing upwards) by the imaging system of the FITS writer. This convention is clearly helpful in the absence of a description of the world coordinates. It does not preclude a program from looking at the axis descriptions and overriding this convention, or preclude the user from requesting a different display. This convention also does not excuse FITS writers from providing complete and correct descriptions of the image coordinates, allowing the user to determine the meaning of the image. The ordering of the image for display is simply a convention of convenience, whereas the coordinates of the pixels are part of the physics of the observation.

5.2. Units in comment fields

If the units of the keyword value are specified in the comment of the header keyword, it is *recommended* that the units string be enclosed in square brackets at the beginning of the comment field, separated from the slash ("`/`") comment field delimiter by a single space character. This widespread, but optional, convention suggests that square brackets should be used in comment fields only for this purpose. Nonetheless, no software should depend on there being units expressed in this fashion within a keyword comment, nor should any software depend on any string within square brackets in a comment field containing a proper units string. This convention is purely for the human reader, although software could be written which would interpret the string only if present and of proper content. If there is an established convention for the units of a keyword, then only those units may be used. An example, using a non-standard keyword, is

```
EXPTIME = 1200. / [s] exposure time in seconds
```

Table 7. Conventional Stokes values.

Value	Symbol	Polarization
1	I	Standard Stokes unpolarized
2	Q	Standard Stokes linear
3	U	Standard Stokes linear
4	V	Standard Stokes circular
-1	RR	Right-right circular
-2	LL	Left-left circular
-3	RL	Right-left cross-circular
-4	LR	Left-right cross-circular
-5	XX	X parallel linear
-6	YY	Y parallel linear
-7	XY	XY cross linear
-8	YX	YX cross linear

5.3. Tables

Binary extension tables (Cotton et al. 1995) use the `NAXIS2` keyword, invented for simple images, to specify the number of rows in a table. It has been suggested that, if the rows of the table are regularly spaced in some world coordinate, that world coordinate could be described with the other axis-2 keywords such as `CTYPE2`*a*, `CRVAL2`*a*, `CDELT2`*a*, and `PC2_2`*a*. Since we know of no software system using this convention with these general keywords, we deprecate the suggestion. There are very powerful and useful general operators such as sorting, editing, and concatenation which alter the value of the row number and therefore corrupt the value of the implied coordinate. If FITS were solely used as an interchange mechanism, these operators would not be relevant. But FITS is now used as the internal format of several software systems for which the general operators are important. Initial row number can be recorded as a column in tables and associated with a physical coordinate via keywords described in Sect. 3.

5.4. Conventional coordinate types

In the first FITS paper, Wells et al. (1981) listed a number of "suggested values" for `CTYPE`*i*. Two of these have the attribute that they can assume only integer coordinate values and that the meaning of these integers is only by convention. These two axis types are in wide-spread use and we wish to repeat their definition here to extend their definitions and to reserve their names and meanings.

The first conventional coordinate is CTYPE*ia* = 'COMPLEX' to specify complex valued data. FITS data are limited to a single real number value at each pixel making this axis necessary to represent data which are weighted complex numbers. Conventional values of 1 for the real part, 2 for the imaginary part, and 3 for a weight (if any) have been widely used.

The second conventional coordinate is CTYPE*ia* = 'STOKES' to specify the polarization of the data. Conventional values, their symbols, and polarizations are given in Table 7.

6. Header construction example

A simple header construction example based on Einstein's Special Theory of Relativity (1905) will serve to illustrate the formalism introduced in this paper. We will construct dual coordinate representations, the first for the rest frame and the second for an observer in uniform motion.

Suppose we have a data cube that, in the rest frame, has the following simple header containing two spatial axes and one temporal axis:

```
 NAXIS = 3,         CTYPE1 = 'X',
NAXIS1 = 2048,      CTYPE2 = 'Y',
NAXIS2 = 2048,      CTYPE3 = 'TIME',
NAXIS3 = 128,       CRVAL1 = 0.0,
CRPIX1 = 1024.5,    CRVAL2 = 0.0,
CRPIX2 = 1024.5,    CRVAL3 = 0.0,
CRPIX3 = 64.5,      CUNIT1 = 'km',
CDELT1 = 3.0,       CUNIT2 = 'km',
CDELT2 = 3.0,       CUNIT3 = 'us',
CDELT3 = 10.0,     WCSNAME = 'Rest frame'.
```

This describes three linear coordinate axes with the reference point in the middle of the data cube.

The spatial and temporal coordinates measured by an observer moving with uniform velocity v in the $+x$ direction are related to the rest coordinates by the Lorentz transformation:

$$\begin{aligned} x' &= \gamma(x - vt)\,, \\ y' &= y\,, \\ t' &= \gamma(t - vx/c^2)\,, \end{aligned}$$

where

$$\gamma = 1/\sqrt{1 - v^2/c^2}\,,$$

and c is the velocity of light. Time in each system is measured from the instant when the origins coincide. From the above header we have

$$\begin{aligned} x &= s_1(p_1 - r_1)\,, \\ y &= s_2(p_2 - r_2)\,, \\ t &= s_3(p_3 - r_3)\,, \end{aligned}$$

where, r_j and s_i are given by CRPIX*j* and CDELT*i*. Thus

$$\begin{aligned} x' &= \gamma s_1(p_1 - r_1) - \gamma v s_3(p_3 - r_3)\,, \\ y' &= s_2(p_2 - r_2)\,, \\ t' &= \gamma s_3(p_3 - r_3) - \gamma v/c^2 s_1(p_1 - r_1)\,. \end{aligned}$$

This set of equations may be rewritten to make the scales, CDELT*i*, the same as the rest frame header:

$$\begin{aligned} x' &= s_1[\gamma(p_1 - r_1) - \gamma v s_3/s_1(p_3 - r_3)]\,, \\ y' &= s_2[(p_2 - r_2)]\,, \\ t' &= s_3[\gamma(p_3 - r_3) - \gamma v/c^2 s_1/s_3(p_1 - r_1)]\,. \end{aligned}$$

Using character "V" as the alternate representation descriptor, a, for the relatively moving frame, we have

$$\begin{array}{ll}
\mathtt{CRPIX1V} = 1024.5, & \mathtt{CTYPE1V} = \mathtt{'X'}, \\
\mathtt{CRPIX2V} = 1024.5, & \mathtt{CTYPE2V} = \mathtt{'Y'}, \\
\mathtt{CRPIX3V} = 64.5, & \mathtt{CTYPE3V} = \mathtt{'TIME'}, \\
\mathtt{PC1_1V} = \gamma, & \mathtt{CRVAL1V} = 0.0, \\
\mathtt{PC1_3V} = -\gamma v/\sigma, & \mathtt{CRVAL2V} = 0.0, \\
\mathtt{PC3_1V} = -\gamma\sigma v/c^2, & \mathtt{CRVAL3V} = 0.0, \\
\mathtt{PC3_3V} = \gamma, & \mathtt{CUNIT1V} = \mathtt{'km'}, \\
\mathtt{CDELT1V} = 3.0, & \mathtt{CUNIT2V} = \mathtt{'km'}, \\
\mathtt{CDELT2V} = 3.0, & \mathtt{CUNIT3V} = \mathtt{'us'}, \\
\mathtt{CDELT3V} = 10.0, & \mathtt{WCSNAMEV} = \mathtt{'Moving\ frame'}.
\end{array}$$

Note that the elements of the PC*i_j* matrix are all dimensionless, $\sigma = s_1/s_3 = 3 \times 10^8 \mathrm{m\,s^{-1}}$ having the dimensions of a velocity. However, in this instance we have seen fit not to apply the strictures of Eq. (4) in normalizing the matrix. In fact, in Minkowski space-time the concept of "distance", on which Eq. (4) relies, differs from the Euclidean norm, the invariant being

$$d = \sqrt{x^2 + y^2 + z^2 - c^2t^2}\,,$$

so one may query the fundamental validity of Eq. (4) in this case. However, the intent of that equation is well served since PC*i_j* and CDELT*i* are divided in a physically meaningful way, especially considering that γ is often close to unity so that PC*i_j* is approximately the unit matrix. Nevertheless, the appearance of the factor γ in each of the elements of PC*i_j* suggests the factorization

$$\begin{array}{ll}
\mathtt{PC1_1V} = 1, & \mathtt{CDELT1V} = 3.0\gamma, \\
\mathtt{PC1_3V} = -v/\sigma, & \mathtt{CDELT2V} = 3.0, \\
\mathtt{PC3_1V} = -v/c^2\sigma, & \mathtt{CDELT3V} = 10.0\gamma, \\
\mathtt{PC3_3V} = 1, &
\end{array}$$

and indeed this does also have a physically meaningful interpretation in that the scales are dilated by the Lorentz-Fitzgerald contraction factor, γ.

7. Summary

The changes to FITS-header keywords are summarized in Table 8. As described in Paper II, for one purpose, CROTA*i* may in some cases be used instead of the new keywords so that the coordinate information may be understood by software systems which have yet to be converted to these new conventions.

Table 8. Coordinate keywords: see also Table 2 for alternate types used in binary tables.

Keyword	Type	Sect.	Use	Status	Comments
WCSAXES*a*	integer	2.2	WCS dimensionality	new	allows WCS specification for degenerate axes, to be explicit rather than implicit.
CRVAL*ia*	floating	2.1.1	value at reference point	extended	meaning of reference point forced by coord. type.
CRPIX*ja*	floating	2.1.1	pixel of reference point	extended	meaning of reference point forced by coord. type.
CDELT*ia*	floating	2.1.1	increment at ref. point	retained	meaning of increment clarified.
CROTA*i*	floating	1	rotation at ref. point	deprecated	replaced by PC*i_j* and CD*i_j*.
CTYPE*ia*	character	2.1.1	coord./algorithm type	extended	Non-linear types have "4-3" form: characters 1–4 specify the coordinate type, character 5 is "-", and characters 6–8 specify an algorithm code for computing the world coordinate value; case dependent.
CUNIT*ia*	character	2.3	units of coord. values	new	case dependent, allowed values and combinations are described in Sect. 4.
PC*i_ja*	floating	2.1.2	transformation matrix	new	linear conversion of pixel number to pixels along coordinate axes; default = 0 $(i \neq j)$, $= 1$ $(i = j)$.
CD*i_ja*	floating	2.1.2	transformation matrix	new	linear conversion of pixel number to relative coordinates; default all 0 if any given, else PC*i_j* applies.
PV*i_ma*	floating	2.1.4	parameter *m*	new	parameters required in some coordinate types; defaults are algorithm-specific; see Papers II and III for usage examples and default conventions
PS*i_ma*	character	2.1.4	parameter *m*	new	parameters required in some coordinate types; defaults are algorithm-specific; see Paper III for usage example and default conventions
WCSNAME*a*	character	2.5	coord. version name	new	optional documentation
CRDER*ia*	floating	2.6	random error	new	uncertainty in coordinate due to random errors; default = 0
CSYER*ia*	floating	2.6	systematic error	new	uncertainty in coordinate due to systematic errors; default = 0

where *j* is a pixel axis number 1 through 99, *i* is an intermediate world coordinate axis number 1 through 99, *m* is a parameter number 0 2 through 99, and *a* is a coordinate description version character blank and A through Z. PC*i_ja* and CD*i_ja* may not occur in the same header data unit.

Acknowledgements. The authors would particularly like to thank Steve Allen (University of California, Lick Observatory) and Patrick Wallace (U.K. Starlink) who provided valuable encouragement, feedback and suggestions over the long period of this work's development. We are grateful for the assistance provided by Bob Hanisch (Space Telescope Science Institute) in bringing the manuscript to its final form. We thank William Pence, Arnold Rots, and Lorella Angelini (NASA Goddard Space Flight Center) for contributing the original text of Sect. 3 and for numerous other suggestions and comments. We also thank Ian George and Lorella Angelini (NASA Goddard Space Flight Center) for contributing the text that formed the basis of Sect. 4 and Francois Ochsenbein (Observatoire Astronomique, Strasbourg) for valuable comments on that and other sections.

The authors also thank the following for comments and suggestions: Lindsey Davis, Brian Glendenning, Doug Mink, Jonathan McDowell, Tim Pearson, Barry Schlesinger, William Thompson, Doug Tody, Francisco Valdes, and Don Wells.

The National Radio Astronomy Observatory is a facility of the (U.S.) National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

The Australia Telescope is funded by the Commonwealth of Australia for operation as a National Facility managed by CSIRO.

References

Adobe Systems, Inc. 1999, *PostScript Language Reference Manual*, Third Edition, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, (see especially pages 293-296)

Calabretta, M. R. & Greisen, E. W. 2002, A&A, 395, 1075, Paper II

Calabretta, M. R. et al. 2003, Representations of distortions in FITS world coordinate systems, in preparation, Paper IV

Cotton, W. D., Tody, D. & Pence, W. D. 1995, A&AS, 113, 159

Einstein, A. 1905, Ann. Physik, 17, 891

George, I. M. & Angelini, L. 1995, Specification of Physical Units within OGIP FITS files, OGIP Memo OGIP/93-001, NASA Goddard Space Flight Center, Greenbelt, Maryland, `http://www.cv.nrao.edu/fits/wcs/OGIP93_001.ps`

Greisen, E. W. & Harten, R. H. 1981, A&AS, 44, 371

Greisen, E. W. 1983, Non-linear Coordinate Systems in AIPS, AIPS Memo No. 27, National Radio Astronomy Observatory, Charlottesville, Virginia, November, `http://www.cv.nrao.edu/fits/wcs/aips27.ps`

Greisen, E. W. 1986, Additional Non-linear Coordinates, AIPS Memo No. 46, National Radio Astronomy Observatory, Charlottesville, Virginia, May 20, `http://www.cv.nrao.edu/fits/wcs/aips46.ps`

Greisen, E. W., Valdes, F. G., Calabretta, M. R. & Allen, S. L. 2003, A&A, in preparation, Paper III

Grosbøl, P., Harten, R. H., Greisen, E. W. & Wells, D. C. 1988, A&AS, 73, 359

Hanisch, R. J., Farris, A., Greisen, E. W., et al. 2001, A&A, 376, 359

Hanisch, R. J. & Wells, D. C. 1988, World Coordinate Systems Representations Within the FITS Format, draft based on notes from a meeting sponsored by the National Aeronautics and Space Administration, Code EZ, held in Charlottesville, Virginia, January 1988. Located at `http://www.cv.nrao.edu/fits/wcs/wcs88.ps.Z`

Harten, R. H., Grosbøl, P., Greisen, E. W., & Wells, D. C. 1988, A&AS, 73, 365

IAU Inf. Bull. 1983, No. 49, 14

McNally, D., ed. 1988, Transactions of the IAU, *Proceedings of the Twentieth General Assembly* (Dordrecht: Kluwer)

Ochsenbein, F., Paul, N. & Kuin, M. 1996, Standards for Astronomical Catalogues, Version 1.5, `http://vizier.u-strasbg.fr/doc/catstd.htx`

Ponz, J. D., Thompson, R. W. & Muñoz, J. R. 1994, A&AS, 105, 53

Wells, D. C., Greisen, E. W. & Harten, R. H. 1981, A&AS, 44, 363