Response to Comments on the Proposed Changes to the FITS Standard Document, version 3.0

Received during the Public Comment Period 23 July 2007 to 22 October 2007, and during the subsequent review by the regional FITS committees.

This document summarizes the public comments that were submitted regarding the proposed new draft version 3.0 of the FITS Standard document. Comments on 43 different topics were submitted as itemized in the following list. The summary of each comment is immediately followed by the technical panel's response. In each case, the response begins with the letter Y or N in parentheses to indicate whether the panel general accepts or not the comment or suggestion. In total, 26 of the comments were accepted and 17 were not accepted.

1 Comments from the Regional FITS Committees

1. Section 2.2, Definition of "FITS file" - To be consistent with section 3.7, should a FITS file be defined as "a format that conforms to the specifications in this OR EARLIER VERSIONS OF THIS standard"?

   Technical Panel's Response: (N) We suspect that this comment refers to the previous draft of the new FITS Standard which proposed a change to the wording in section 3.7. That proposed change was withdrawn in the final draft, therefore we see no need to make any change to the definition of "FITS file".

2. Section 2.2, Definition of "Primary Header" - A picky point, but it may not be clear from the sentence structure whether the added ",if present" refers to the first header or the primary data array. It could be interpreted to mean the primary header is optional.

   Technical Panel's Response: (Y) The final clause in that sentence is now enclosed in parentheses to clarify that the "if present" refers only to the data array and not to the header.

3. Section 4.3.2 - Last sentence - If a specific unit is "recommended" by the standard, how can you say it "must" be used?

   Technical Panel's Response: (Y) The wording of that sentence has been revised to remove this conflict, and the sentence has been moved to the 2nd paragraph in section 4.3 where it is more relevant.

4. Section 4.4.2.1 Date Keyword - remove second "a" in first sentence last paragraph (verbatim copy of a another HDU?).

   Technical Panel's Response: (Y) This typographical error has been fixed.

5. Section 4.4.2.2 Equinox - Why should an EQUINOX value of 2000 imply ICRS? How does one indicate FK4 or FK5 coordinates that were precessed to year 2000?

   Technical Panel's Response: (Y) This definition of EQUINOX is incorrect. This keyword is correctly defined in section 8.3, therefore the definition in section 4.4.2.2 has been deleted.

In addition, the definition of the deprecated EPOCH keyword has also been moved to section 8.3 and added to table 8.3.

2 General Comments

1. Italicizing [*must*, *should*, *recommend*, etc].*may* be distracting. *Must* it be done? I *should* not *recommend* it. *Shall* it be *optional*? (Rob Seaman)

   Technical Panel's Response: (N) Italicizing these special words is intended to make the standard clearer and more precise by alerting the reader that these words have special significance as defined in RFC 2119. It appears that most other reviewers of this draft did not object to this typographical emphasis, therefore we have not removed it.

2. *Any chance of blessing checksums on this go-around?* (Rob Seaman)

   Technical Panel's Response: (N) This would go beyond the purpose of the current exercise which is to just update and clarify the FITS Standard.

3. *There is a very real possibility that UTC will be turned into something other than a flavor of Universal Time, i.e., other than an approximation to Greenwich Mean Time. Perhaps we should consider how best to deal with such a situation. [...] This is a broader issue than FITS, but we would have to start somewhere.* (Rob Seaman)

   Technical Panel's Response: (N) This issue is neither simple nor sufficiently well defined to be addressed by this panel at this time.

4. *I suggest introducing "keyvalue", "keyrecord", "keyfield" as contractions for "keyword value", etc.* (Mark Calabretta)

   Technical Panel's Response: (N) Introducing new short-hand terms could cause confusion, especially for readers whose native language is not English, and does not seem warranted in this case. For the record, "keyword value" , "keyword record" and "keyword field" appear 10, 19 and 1 times, respectively, in the document.

5. *It would be useful for the FITS 3.0 document to list inconsistencies in syntax [between previous versions of the Standard] in an appendix, e.g. such as the requirement for a slash between value and comment introduced at version 2.0.* (Mark Calabretta)

   Technical Panel's Response: (N) The panel has declined to do this because it would require a significant effort to compile a list of all the changes in the previous versions of the FITS Standard and because the resulting list would primarily only be of historical interest.

6. *We need a "Don't Make Me Think" policy with the FITS format. As a programmer, even after reading the documentation on the FITS format, I still struggle with the header portion of the file. How does one know how many 2880 byte chunks precede the image data???? Like the ARF files we use currently at my work (defense) the format is overly complex and poorly thought out.* (Kevin Thomas)

   That question is answered in the docs. [...] There weren't any other questions in the original post, so it's hard to provide any other answers other than to note that FITS was designed *when 32 k was a lot of directly addressable memory, and everything else was found by running the tape back and forth until the item in question was either found or not.* (Steve Allen)

More standards might benefit from users and programmers having to think. [...] More standards could benefit from a similar dose of self-restraint in minimizing resource requirements. (Rob Seaman)

Technical Panel's Response: (N) The specific question about the size of the header is answered in section 3.3.1 and also in the definition of the END keyword in section 4.4.1.1. The more general implication that the Standard is not easy to read is perhaps justified, but the Standard is intended as a precise, legalistic definition of the rules of FITS. There are a number of other overviews of the FITS format (e.g., the FITS Users Guide, the Wikipedia article on FITS, and several other introductory FITS documents available from the FITS Support Office web site) what are more appropriate for new users of FITS.

7. *I assert that despite its high google score, "two's complement" has an inappropriate apostrophe.* (anonymous)

   Technical Panel's Response: (N) The rational for this punctuation is that it means "complement of two". In any case, searches of the Internet show that this form is more widely used in other publications than any of the alternative forms.

3 Comments on Specific Sections

- Section 2.2, on the definition of "deprecate":

  *It is perhaps a little confusing to refer to "applications" in this context, when the standard governs data files and their interpretation, but only advises the functionality of applicable software. The following wording is an attempt to clarify the definition: "This term refers to obsolete features or structures in FITS that *should not* be used in new FITS files, but *shall* remain valid indefinitely. FITS users and data engineers should be aware that the use of deprecated structures is considered bad form and should be avoided."* (Dick Shaw)

  [Dick Shaw's] wording is a great improvement but still it doesn't quite capture the dichotomy between FITS writers and FITS readers, new and old. There are two types of deprecation:

  1. *Constructs that ought never be used by writers (but must still be understood by readers) e.g. omitting the "/" comment delimiter, or implicit decimal points in ASCII table fields. It is safe to outlaw such usage because doing so has no effect on old readers.*
  2. *Constructs that shouldn't be used by writers but might still be in order to help old readers (and must still be understood by new readers). EXTEND, EPOCH and CROTAn are examples. (Mark Calabretta)*

  Technical Panel's Response: (Y, partially) The panel has clarified the use of the term "deprecate" throughout the Standard and the references to "applications" within this context have been deleted.

- Section 3.5, which deprecates special records

I think it's okay to deprecate 'special records' as the general extensions should be able to handle our future needs. In an age with computer virus, it may be safer to have some handle on what can be in a FITS file. (Preben Grosbol)

*The justification given is appropriate. The special records rule was the biggest single escape hatch in the original FITS Agreement of 1979; it was the basis for the Generalized Extensions Agreement of 1984, which led to TABLE, BINTABLE and IMAGE extensions. I agree that the Generalized Extensions Agreement "appears to be sufficient to meet most future needs". Indeed, I would go further and say that the burden of proof is on anybody who claims that the 1984 Agreement is not able to satisfy some astronomical data structure requirement. The variety of conventions that have been built on top of BINTABLE show that it alone has enough versatility and functionality to solve an enormous range of problems. Also, if we encounter something that BINTABLE cannot do, it is always possible for us to encapsulate *any* other data format within FITS, if the other format solves a problem that FITS can't solve, by creating extensions with XTENSION='other format' and the other file in an NAXIS=1 byte array. Therefore, although this item makes me nervous, I have decided to accept deprecation of special records.* (Don Wells)

Extensions ARE special records from the point of view of classic FITS readers. Rather than deprecating or attempting to ban them, perhaps just explicitly state that new uses for special records require IAUFWG approval. (Rob Seaman)

Technical Panel's Response: (Y) Although extensions may have been considered to be special records when they were first developed, the Standard now makes it clear in section 3.5 that conforming extensions are not special records. We agree that rather than "deprecate" special records (i.e. 'discourage' any further use), it is better to make the stronger statement that any use of special records is "restricted" and requires IAU FWG approval.

- Section 3.7, which adds a new rule that exempts existing FITS files from any new FITS requirements

  [This rule is] unenforceable in the absence of some explicit mechanism tagged to the DATE keyword (for instance). On the other hand, such a mechanism would be equivalent to explicitly versioning each file. (Rob Seaman)

  The [proposed new rule] is understandable but the implications [are] very wide. It would allow dramatic changes such as redefinition of keywords. Something like this is needed but we may need to consider the best wording. (Preben Grosbol)

  In principle the new Sec. 3.7 introduces a FITS version that is in FITS v3+ certain things are not valid any more. We previously were very reluctant to introduce the concept of versioning but maybe it cannot be avoided. It is a rather principle matter which should be discussed in some detail. (Preben Grosbol)

  The suggested new wording of 3.7 can't serve as a substitute for an explicit version since A) not all files have DATE keywords, so how would you know, and B) it will take several years - perhaps never - for all prior software to adopt the new standards. (Rob Seaman)

  Technical Panel's Response: (Y) The panel withdraws this recommendation. Since this modification would have little or no practical implications for FITS file writers, and given the amount of controversy that it has generated over the issue of FITS "versions", this proposed change has been deleted.

- Section 4.1.1, which recommends that the keyword order be preserved during data processing operations

  Harmless, but also unenforceable. Not just an attempted constraint on software, but also on humans editing headers manually using IRAF hedit, for instance. (Rob Seaman)

  Technical Panel's Response: (N) The panel sees no harm in adding this recommendation, especially since applications may ignore it for any reason.

- Section 4.1.2.3, which add a new requirement that keywords with a value must not appear more than once in a header

  There are many headers that unintentionally repeat keywords. They should not therefore be rendered unconforming FITS. (Rob Seaman)

  It may be too strong to forbid such repeats of keywords. It should certainly be deprecated but it may place a significant load on many applications which just want to add some keywords to a header. They would be required to actually check if they would duplicate or not. A deprecation and definition that the last value takes precedence may be more appropriate. (Preben Grosbol)

  I have many examples (hundreds of thousands?) of files in which keywords are repeated. Rather than the wording in the current proposal, I would replace the attempt at a requirement with a strong recommendation and a clarification that the final copy of any such repeated keyword should take precedence. (Rob Seaman)

  I strongly support [Rob Seaman's statement]. The proposed text makes such files invalid FITS, retrospectively. Taking the last instance of a keyword is a much more reasonable interpretation. (Tim Pearson)

  I similarly cannot see the value of this particular proposed change: FITS readers will need to support repeated keywords forever, given the very large numbers of existing files with them, so it's not even as if this would simplify reading FITS. I am also very much in favour of instead simply clarifying that the last occurrence has precedence. (Thierry Forveille)

  I agree with Thierry that there are many files which have repeated keywords, but I agree with another poster that there are existing implementations which assume it's the first instance, not the last instance, which prevails. So I think we should just strongly deprecate (not ban, and not impose an interpretation). (Jonathan McDowell)

  *I'll just second this. I have seen many cases where a FITS header gets multiple keyword instances after it has been modified by different programs. Software may or may not use the last instance as a runtime value; We should strongly recommend that software which writes to a header not create multiple values. If there already *are* multiple values, frankly it is not often clear what to do, as one does not like to delete information from a header.* (Doug Tody)

  *One approach would be to say that headers *should not* contain repeated keywords, and if a repeat does occur then the value is not defined (unless the values are identical).* (Dick Shaw)

  Would it be too complex to make the non-duplication of reserved words mandatory and of others just strongly recommended? (Peter Bunclark)

  All cases I have seen were instances of "updating" the information by appending a second copy of the keyword an the (then) end of the header. It is easier to write at the end of the header, so I'd expect the latest version of the keyword to almost always be last in the header. (Thierry Forveille)

Technical Panel's Response: (Y) The panel accepts that there are too many FITS files in existence with duplicated keywords to simply forbid this practice in a valid FITS file. Instead, the wording of this section has been modified to state that

- all the mandatory FITS keywords defined in the Standard must not appear more than once in any header, and
- all other keywords that have a value should not appear more then once in any header, and
- if a keyword with a value appears more than once with different values, than the value is indeterminate.

It is not practical at this point to make any recommendation in the Standard as to which instance of a duplicated keyword should take precedence; existing software systems have different behaviors with some returning the first occurrence, while other systems return the last occurrence, and while still others return the next occurrence after the most recently read keyword.

- Section 4.2.1: *I don't see the point of distinguishing between "fixed-format" and "free-format" character strings. Fixed-format is just a special case of free-format and the term is never used elsewhere, not even to describe the special formatting required for the XTENSION keyword. The same comment applies for integer and floating-point values.* (Mark Calabretta)

  Technical Panel's Response: (N) The Standard requires that the mandatory FITS keywords be written in fixed format to facilitate the decoding of FITS files on even the most basic computer systems. The panel sees no compelling reason to change this requirement now, especially since it is likely that some existing FITS readers depend on the value of the XTENSION, BITPIX and NAXIS keywords to be in fixed format. Also, while it is a matter of personal preference, the use of the fixed format for the keywords tends to improve the 'neatness' and 'legibility' of listings of the header contents.

- Section 4.4.1.1, which recommends that BITPIX data type should be appropriate to the range and accuracy of the data

  Few users will choose 64 bits when 16 will do. Those that do may well have a good reason. (Rob Seaman)

  Technical Panel's Response: (N) While we agree that most users will naturally choose an appropriate data type, it does no harm to make this an explicit recommendation. The wording of this is very similar to the existing recommendation on the choice of binary table column data formats in section 7.3.3.1 that the IAU FITS Working Group inserted when it added support for 64-bit integers to FITS.

- Section 4.4.1.1 *The ordering of NAXISn is never explicitly restricted to increasing numerical order. The only statement for any of the mandatory keywords is presented in table 4.5, which suggests NAXISn be ordered, but never outright says it.* (Rob Seaman)

  Technical Panel's Response: (Y) An explicit statement that the NAXISn keyword must appear in order of increasing n has been added. Similar statements have been added in sections 6.1.1 (random groups) and 7.1.1 (image extensions). In addition, explicit statements that the TFORMn and TBCOLn keywords "must appear for all values of n = 1, ..., TFIELDS and for no other values of n" have been added in the definitions of these table extension keywords.

- Section 4.4.1.2, which *requires* XTENSION names be registered with the IAU FWG

  This is a requirement placed by the IAU, not FITS per se. Might such policies better be concentrated in a common section? (Rob Seaman)

  I would prefer a strong urging accompanied by the usual justification rather than a 'must', precisely because XTENSION is our most important escape hatch, and I don't want to inhibit experimentation unnecessarily. (Don Wells)

  Technical Panel's Response: (N) The requirement to register new extension types with the IAUFWG is not new and has been in all previous versions of the standard (in section 3.4.1.1). The only change we are proposing is to repeat this requirement for emphasis within the description of the XTENSION keyword in section 4.4.1.2.

- Section 4.4.1.2, which requires that GCOUNT immediately follow PCOUNT in all conforming extensions

  In principle okay, but one should check if this would require existing software to be changed. There may still be legacy tasks which write e.g. Random groups format. (Preben Grosbol)

  I guess I'd like to know if there are any such extensions. If not, this is relatively safe. If so, make it a strong recommendation for an explicit list of grandfathered extension types and an absolute requirement for any newly defined extensions. (Rob Seaman)

  At least some of the FOREIGN extensions generated at NOAO have the order of these 2 keywords reversed. (William Pence)

  We'll look into the [reversed keywords in FOREIGN extensions] that [Pence] described. I would expect most extension types, including FOREIGN, to be conformable to this more strict keyword ordering whether it is required or merely preferred. (Rob Seaman)

  Technical Panel's Response: (Y) This new requirement only affects conforming extensions and is not relevant to random groups files which may have the PCOUNT and GCOUNT keywords in either order. This specific keyword order has always been required in all standard extensions (image extensions, ASCII table extensions, and binary table extensions), so this new requirement only affects the small number of other conforming extensions. While there is one known case of "FOREIGN" extensions produced at KPNO that have the opposite order of these 2 keywords, these files are mainly intended for internal use and have not been widely distributed outside KPNO. The definition of the FOREIGN extension in the Registry of FITS conventions now requires this specific keyword order.

- Section 4.4.2.1, which makes the EXTEND keyword optional in the primary header of FITS files that have extensions

  I am aware of several IDL FITS libraries which follow the existing standard and require the presence of the EXTEND keyword before extensions will be examined. (Craig Markwardt)

  The catfits task in the STSDAS package under IRAF prints a "table of contents" for a FITS file. If the EXTEND keyword is absent, catfits does not look for extensions. (Phil Hodge)

  At the time when extensions were introduced, most readers would not even think of looking for them. Thus, it was important that users could look on their primary header and see (and then complain) that there were extension data available in the file which were not detected by the local reader. Now that extensions are common and most readers would check, I see little *problem in making EXTEND optional. The ESO readers, I know of, all would scan the full file no matter what.* (Preben Grosbol)

The "fault" about EXTEND was in its original definition. The keyword EXTEND=T was present "if the FITS file MAY contain extensions" and it being equal to T did "not imply extensions be present". But it was/is pretty useless to tell whether the file really contains extensions. So the safest way taken by FITS readers was to scan anyhow after the PHDU for named XTENSION kwds. (Lucio Chiappetti)

I don't particularly object to the EXTEND change (which doesn't invalidate any old FITS file, just gives a (very) small added flexibility to new ones, so doesn't break the "once FITS always FITS rule"), but don't see its point either. Of course it saves 80 bytes in every FITS file, but I suppose there must be some stronger reason. (Thierry Forveille)

Technical Panel's Response: (N) Making the EXTEND keyword optional is intended to rationalize and simplify the rules defining the FITS format. This keyword is no longer useful and it can potentially lead to confusion if the keyword is absent in FITS files that actually do have extensions present. Since the existence of extensions can be determined simply by reading the FITS file, we recommend that this keyword be made optional. While this could have an adverse impact on a some software, as was noted above by a few of the reviewers, we believe the negative impact of this change is fairly small and is out weighted by the positive benefit of simplifying the rules that all FITS writers must follow.

- Section 4.4.2.1, which states that the EXTEND keyword can only have a value = T.

  The SECCHI instrument on the STEREO mission produces some files with table extensions, and other files which don't. The value of the EXTEND keyword is set to either T or F to account for this. (William Thompson)

  Technical Panel's Response: (Y) This section has been reworded to not exclude EXTEND = F.

- Section 4.4.2.1 BLOCKED keyword

  Following the discussion of the FITS 'version' issue (see section 3.7 above) it is inconsistent to state that the BLOCKED keyword "shall not" be used. This implies that it became illegal to to use this keyword, starting with the publication of some version of the Standard. It would be more consistent with the meaning of the word 'deprecate' to state that this keyword 'should not' be used rather than 'shall not'. The same applies to the definition of the EPOCH keyword in section 4.4.2.2.

  Technical Panel's Response: (Y) The description of these 2 keywords has been changed to 'should not be used' rather than 'shall not be used'.

- Section 4.4.2.3 REFERENC keyword

  The ADS ID is recommended to be used in the REFERENC keyword. Recently I more and more noticed the use of DOI numbers for stuff like this (see http://doi.org/). As this seems to be standardized across many more disciplines than just Physics and Astronomy, should this not be the recommended way to store a reference? Or perhaps recommend this in addition to the ADS reference? (Peter Weilbacher)

  Technical Panel's Response: (Y) This section has been modified to recommend either DOI numbers or the ADS reference string.

- Table 4.9 and 7.7 *It would look nicer to start the tables with BITPIX 8* (Preben Grosbol)

  Technical Panel's Response: (Y) The tables have been reordered.

- Section 4.4.2.5: *Since the description of WCS keywords like CDELTn was moved to section 8, it would be good to have a reference to that section in section 4.* (Preben Grosbol)

  Technical Panel's Response: (Y) A cross reference to section 8 has been added.

- Section 4.4.3, which removes the previous restriction that no keyword in the primary header can specify the presence of a specific extension in the file.

  As there are many recommendations in the document, it would be good to retain a deprecation of explicit reference to other extensions. The point is broken links if HDU's are moved. This opens the old issue on how to create a unique reference to any HDU. It would be an advantage for some applications, such as in interferometry, where cross-references between binary table sometimes are used. (Preben Grosbol)

  Technical Panel's Response: (N) In the absence of any other recommended way to provide cross references between different FITS HDU, the panel does not think it should deprecate specific references from one HDU to another. While the links may break, this sort of cross referencing mechanism may be useful in some circumstances. This section has been reworded to advise that such keyword be used with caution.

- Section 7.2.2 and 7.3.2 TTYPEn keyword

  The reason for avoiding '-' was the mapping of the TTYPEn names into variables is some languages. It is irritating to have '-' in keywords for the same reason and most applications would just substitute them with '_'. I would either retain the old wording or if '-' should be allow at the same point deprecate the usage of it. (Preben Grosbol)

  Technical Panel's Response: (Y) The proposed change to add the hyphen character to the list of recommended characters in table column names has been removed. The text has been expanded to explain why the other ASCII text characters should be avoided.

- Section 7.2.2 *It may be good to repeat a reference to fortran in the description of the TDISPn keyword.* (Preben Grosbol)

  Technical Panel's Response: (Y) The previous technical panel in 1998 deliberately removed most references to Fortran from the document, but we agree that in this case it is appropriate to mention that the TDISPn keyword have the same interpretation as the fortran format codes.

- Section 7.2.5, which deprecates implicit decimal points in numeric fields of an ASCII table

  We should check it with the data centers. The reason for including such cases was to accommodate direct encoding of legacy tables which could have used such formats to save space (at the time when a punch cards was real and only had 80 columns). (Preben Grosbol)

  Technical Panel's Response: (Y) We checked that the VizieR and HEASARC archives do not have any known cases of this usage. We recommend only deprecating this usage, rather than totally banning it, because a) the original FITS paper that defined ASCII tables specifically mentions this usage, and b) because many existing software systems already support this and would read the correct value if an ASCII table were to use this feature.

- Section 7.2.5, which forbids embedded spaces in numeric fields of an ASCII table

  Are there any examples of such usage in the field? If there are no known instances, "outlawing" is equivalent to clarifying the standard. (Rob Seaman)

  Technical Panel's Response: (Y) There are no cases of this usage as far as we know (we specifically checked with the maintainers of the the VizieR and HEASARC archives). If there were any such cases, it is likely that current software systems may silently read an incorrect value or would exit with an error. Given the potential severe consequences, it seems appropriate to totally outlaw this usage rather than simply not recommending or deprecating it.

- Section 7.3.3.2, which removes the option to use the heap and the PCOUNT keyword in ways other than described in the variable length array section 7.3.5

  This wording makes me nervous because tricks with the heap and the gap area are an obvious potential escape hatch that we left in the Binary Tables Agreement in 1991. I decided to examine the color-coded differences document (very nice!). I see changes marked that seem to be related to this item, but I am unsure which changes accomplish the goal of the item. Perhaps this item in the Recommended Changes document should have some text added to clarify this point, or maybe the differences document is missing some color-coding. (Don Wells)

  Technical Panel's Response: (N) The recommendation is to make a subtle change the wording of section 7.3.3.2, from this: "One use for this data area is described in section 7.3.5" to this: "The use of this data area is described in section 7.3.5". The current ambivalent wording (i.e. "One use" instead of "The use") is left over from when the variable length array convention had not yet been officially approved and was only described in an appendix to the Standard. Now that this convention has been approved (in 2005), it is appropriate to state more definitely how the heap is intended to be used. This is consistent with all the other FITS data structures (e.g. the primary array, or random groups) that are defined to be used in a single specific way. The technical panel also recommends deleting the final sentence of this paragraph ("This does not preclude other uses for these bytes.") for a similar reason. This boilerplate statement was routinely added at the end of all the conventions that were described in the unofficial appendices in the FITS standard. Now that this convention has been officially moved into the body of this standard, this disclaimer is not necessary nor strictly appropriate.

- Section 7.3.5, describing the variable length array convention

  *I encountered two sentences that seem to me to be inconsistent. The first is "The meaning of a negative value for either of these integers [in the array descriptor] is not defined by this standard." The second is "The storage referenced by an array descriptor must lie entirely within the heap area; negative offsets are not permitted." The first sentence leaves open the option of a negative offset (and that is an obvious escape hatch, because the integers are definitely *signed*), but the second sentence closes off the option. (The text of the second sentence is present in v2.1, so this isn't the change made for item 21; I don't remember when this prohibition was introduced; certainly the IAU-FWG could remove it if they wish.) Apparently the second sentence overrules the first sentence, but the situation makes me nervous. Negative offsets would address the gap area, and this might be the basis for some clever design that I cannot imagine at this moment. However, I see nothing to stop a designer from using* *simple integer fields in the binary table to contain pointers that her software would construe as data descriptors for the gap area. I have decided not to request removal of the negative offset prohibition, but I do suggest that the apparent conflict of the two sentences be reviewed.* (Don Wells)

*I'm really also concerned by this inconsistency raised by Don and I did ask the same question before. I don't think it's wise to try to allow heap offset to be larger than 2**31 – if the offset is a signed integer, then the limitation of the heap offset should be 2**31. If more is wished, then move to 64-bit integers. I'm sure the gain in space is minimal, just the rare cases when the heap size is between 2**31 and 2**32 would diminish the size of the pointers. And 32-bit machines won't anyway be able to use heaps larger than 2**31 anyway.* (Francois Ochsenbein)

*And now that we have P and Q pointers, if one needs, wants and can go beyond 2**31, one has just to use Q. I thought this issue was settled long ago.* (Lucio Chiappetti)

Technical Panel's Response: (N) The topic of whether the heap address is a signed or unsigned integer and the meaning, if any, of negative values was the subject of a extensive debate by the IAU FITS Working Group in 2005 when the variable length array convention was approved as part of the standard. Without specific direction from the IAU FWG, this panel does not wish to reopen this debate.

We simply point out that the 2 statements cited above by Don Wells are not necessarily contradictory. Negative descriptor values could be allowed if one adopts a convention which maps those values into a positive heap offset. This is not unprecedented in FITS; the BITPIX keyword is another example where one must apply a simple mapping function to the negative keyword value to derive the positive number of bits per pixel.

- Section 8: Various corrections to the WCS description (Mark Calabretta)]
  - *where CROTA2 is mentioned it should be stated in this context that, while it may occur with CDi_ja, it must not occur with PCi_ja.*
  - *In table 8.1 and elsewhere in the related text, the 'a' symbol used for the alternate code should be in math font.*
  - *Table 8.2: it would be better if RESTFREQ was presented in the same way as RADECSYS. A footnote describing these forms would be useful.*
  - *Table 8.2 The note at the bottom reverses the meaning of i and j. In that note, k also refers to column number in a pixel list.*
  - *In section 8.2.1, there are a number of keywords, ijPCna, ijCDna, iVn_ma, and so on, where 'a' could be pushed right out of the 8-char keyword name for plausible values of i, j, k, n, and m. In such cases 'a' is still said to be "blank" although it is not the blank character. (This isn't really made clear in Paper I.)*
  - *Section 8.3 probably should indicate that no time system (UTC, TAI, TCB, etc.) is currently defined for MJD-OBS or any of the other time- related keywords. (This is not important for anything in the existing WCS papers.)*
  - *Section 8.4, definition of RESTWAV: Misspelling of "Vaccuum"*
  - *Table 8.6 the footnote mark should be given in the caption as in table 8.5.*
  - *Section 8.4.1 the reference to Table 8.6 should be 8.7.*

- *Section 8.4.1, footnote regarding VELOSYS keyword: how could CUNITia (not CUNITka) substitute for VELOSYSa anyway? Or should the footnote really refer to the units of VELOSYSa specified later?*
- *Section 8.4 1 in the description of ZSOURCEa, the reference to ZSOURCEa should be SSYSSRCa.*
- *there are a few anomalies in the set of WCS keywords as published [that should be corrected]: The main one is that no bintable or pixel list equivalent of DATE-OBS is defined. Also, the pixel list form for WCSNAMEa was mistakenly specified as TWCSna instead of WCSNna.*

Technical Panel's Response: (Y) These corrections have been applied to the revised draft.

- Section 8.2 CDELTn keyword

  [There was an extended discussion about whether the CDELTn keyword could have a value = 0, which concluded that it could not.]

  Technical Panel's Response: (Y) The definition of CDELTn has been revised to state that the value must be non-zero.

- Section 8.3 *Is keyword "RADECSYS" an allowed alias for "RADESYS" for the primary WCS? I didn't see that option in WCS Paper II.* (Phil Hodge)

  Technical Panel's Response: (Y) RADECSYS (and RESTFREQ) are noted as deprecated forms of the standard keyword, following a recommendation from Mark Calabretta.

- Section 8.4.1, on spectral coordinate reference frames

  the discussion at the bottom should indicate that the diurnal Doppler correction is only weakly dependent on position and therefore great accuracy is not required. (Mark Calabretta)

  This depends on the context, and on what's meant by "no great accuracy": it's 1 m/s for 7 arcsec (worst case position on the sky), so for planet searches we need arcsecond precision or so (our best precision is under 1 m/s, and we want some safety margin to keep that particular contribution to the error budget negligible). Perhaps just give the 7" == 1 m/s equivalence here and let readers make their own decision? (Thierry Forveille)

  In the context of the discussion on the bottom of p92, by position I meant location on Earth, not direction towards the source. By "great accuracy is not required" I meant accuracy better than a few metres! On the Earth's surface 1m = 32mas, which ought to be good enough even for planet hunters. (Mark Calabretta)

  Technical Panel's Response: (Y) The discussion of the diurnal Doppler shift has been modified to note that the correction is only weakly depended on the position on the earth.

- Section 8 *How does one write an array of complex number to a primary array? This does not seem to be mentioned anywhere in the standard.* (Randall Wayth)

  Technical Panel's Response: (Y) A new section 8.5 has been added to the WCS chapter to describe the 2 "conventional" axis types, i.e., for complex numbers and stokes parameters. The wording for this section is taken from the first WCS paper.

- Appendix A, *It would be better to name Appendix A 'Formal Syntax of Keyword Records' to be in line with section 4.1* (Preben Grosbol)

  Technical Panel's Response: (Y) The name has been shortened, to "Syntax of Keyword Records" so that it fits on a single line of text.

- Appendix A, Formal Syntax of Keywords

  Why not use BNF notation (c.f. wikipedia)? Also, there appear to be anomalies in some of the definitions [...] (Mark Calabretta)

  Technical Panel's Response: (N) This appendix uses a simplified variation of BNF notation. Under the notation as defined here, the definitions are correct.

- Appendix F, Reserved Extension Type Names: *It was recently learned that the Hinode satellite project is distributing FITS files with an extension named "DUMP".*

  Technical Panel's Response: (Y) Substantial changes have been made to this appendix, to document the existence of the DUMP extension. Also, COMPRESS, FITS, and FILEMARK have been removed from the list of reserved names because they have never been used.