4.4.2.6. Extension keywords

The next three keywords were originally defined for use within the header of a conforming extension, however they also *may* appear in the primary header with an analogous meaning. If these keywords are present, it is *recommended* that they have a unique combination of values in each HDU of the *FITS* file.

`EXTNAME` keyword. The value field *shall* contain a character string to be used to distinguish among different extensions of the same type, i.e., with the same value of `XTENSION`, in a *FITS* file. Within this context, the primary array *should* be considered as equivalent to an `IMAGE` extension.

`EXTVER` keyword. The value field *shall* contain an integer to be used to distinguish among different extensions in a *FITS* file with the same type and name, i.e., the same values for `XTENSION` and `EXTNAME`. The values need not start with 1 for the first extension with a particular value of `EXTNAME` and need not be in sequence for subsequent values. If the `EXTVER` keyword is absent, the file *should* be treated as if the value were 1.

`EXTLEVEL` keyword. The value field *shall* contain an integer specifying the level in a hierarchy of extension levels of the extension header containing it. The value *shall* be 1 for the highest level; levels with a higher value of this keyword *shall* be subordinate to levels with a lower value. If the `EXTLEVEL` keyword is absent, the file *should* be treated as if the value were 1.

The following keyword is optional, but is *reserved* for use by the convention described in Appendix K. If present it *shall* appear in the extension header immediately after the mandatory keywords, and be used as described in the Appendix.

`INHERIT` keyword. The value field *shall* contain a logical value of `T` or `F` to indicate whether or not the current extension should inherit the keywords in the primary header of the FITS file.

4.4.2.7 Data Integrity Keywords

The two keywords described here provide an integrity check on the information contained in *FITS* HDUs.

`DATASUM` Keyword. The value field of the `DATASUM` keyword *shall* consist of a character string that *should* contain the unsigned integer value of the 32-bit 1's complement checksum of the data records in the HDU (i.e., excluding the header records). For this purpose, each 2880-byte *FITS* logical record should be interpreted as consisting of 720 32-bit unsigned integers. The 4 bytes in each integer *must* be interpreted in order of decreasing significance where the most significant byte is first, and the least significant byte is last. Accumulate the sum of these integers using 1's complement arithmetic in which any overflow of the most significant bit is propagated back into the least significant bit of the sum.

The `DATASUM` value is expressed as a character string (i.e., enclosed in single quote characters) because support for the full range of 32-bit unsigned integer keyword values is problematic in some software systems. This string may be padded with non-significant leading or trailing blank characters or leading zeros. A string containing only one or more consecutive ASCII blanks may be used to represent an undefined or unknown value for the `DATASUM` keyword. The `DATASUM` keyword may be omitted in HDUs that have no data records, but it is preferable to include the keyword with a value of 0. Otherwise, a missing `DATASUM` keyword asserts no knowledge of the checksum of the data records. Recording in the comment field the ISO-8601-formatted Datetime when the value of this keyword record is created or updated is *recommended*.

`CHECKSUM` Keyword. The value field of the `CHECKSUM` keyword *shall* consist of an ASCII character string whose value forces the 32-bit 1's complement checksum accumulated over the entire *FITS* HDU to equal negative 0. (Note that 1's complement arithmetic has both positive and negative zero elements). It is *recommended* that the particular 16-character string generated by the algorithm described in Appendix J be used. A string containing only one or more consecutive ASCII blanks may be used to represent an undefined or unknown value for the `CHECKSUM` keyword.

The `CHECKSUM` keyword value *must* be expressed in fixed format, when the algorithm in Appendix J is used, otherwise the usage of fixed format is *recommended*. Recording in the comment field the ISO-8601-formatted Datetime when the value of this keyword record is created or updated is *recommended*.

If the `CHECKSUM` keyword exists in the header of the HDU and the accumulated checksum is not equal to -0, or if the `DATASUM` keyword exists in the header of the HDU and its value does not match the data checksum then this provides a strong indication that the content of the HDU has changed subsequent to the time that the respective keyword value was computed. Such an invalid checksum may indicate corruption during a prior file copy or transfer operation, or a corruption of the physical media on which the file was stored. It may alternatively reflect an intentional change to the data file by subsequent data processing if the `CHECKSUM` value was not also updated.

Normally both keywords will be present in the header if either is present, but this is not required. These keywords apply *only* to the HDU in which they are contained. If these keywords are written in one HDU of a multi-HDU *FITS* file then it is *strongly recommended* that they also be written to every other HDU in the file with values appropriate to each HDU in turn; in that case the checksum accumulated over the entire file will equal -0 as well. The `DATASUM` keyword *must* be updated before the `CHECKSUM` keyword. In general updating the two checksum keywords should be the final step of any update to either data or header records in a FITS HDU. It is highly *recommended* that if

Table 21: WCS and celestial coordinates notation.

Variable(s)	Meaning	Related FITS keywords
i	Index variable for world coordinates	
j	Index variable for pixel coordinates	
a	Alternative WCS version code	
p_j	Pixel coordinates	
r_j	Reference pixel coordinates	CRPIX*ja*
m_{ij}	Linear transformation matrix	CD*i*_*ja* or PC*i*_*ja*
s_i	Coordinate scales	CDELT*ia*
(x, y)	Projection plane coordinates	
(ϕ, θ)	Native longitude and latitude	
(α, δ)	Celestial longitude and latitude	
(ϕ_0, θ_0)	Native longitude and latitude of the fiducial point	PV*i*_1*a*† , PV*i*_2*a*†
(α_0, δ_0)	Celestial longitude and latitude of the fiducial point	CRVAL*ia*
(α_p, δ_p)	Celestial longitude and latitude of the native pole	
(ϕ_p, θ_p)	Native longitude and latitude of the celestial pole	LONPOLE*a* (=PV*i*_3*a*†), LATPOLE*a* (=PV*i*_4*a*†)

Notes. † Associated with *longitude* axis *i*.

in the CRPIX*i* keywords, and the world coordinates at the reference point are encoded in the CRVAL*i* keywords. For additional details, see Greisen & Calabretta (2002).

The third step of the process, computing the final world coordinates, depends on the type of coordinate system, which is indicated with the value of the CTYPE*i* keyword. For some simple, linear cases an appropriate choice of normalization for the scale factors allows the world coordinates to be taken directly (or by applying a constant offset) from the x_i (e.g., some spectra). In other cases it is more complicated, and may require the application of some non-linear algorithm (e.g., a projection, as for celestial coordinates), which may require the specification of additional parameters. Where necessary, numeric parameter values for non-linear algorithms *must* be specified via PV*i*_*m* keywords and character-valued parameters will be specified via PS*i*_*m* keywords, where *m* is the parameter number.

The application of these formalisms to coordinate systems of interest is discussed in the following sub-sections: Sect. 8.2 describes general WCS representations (see Greisen & Calabretta 2002), Sect. 8.3 describes celestial coordinate systems (see Calabretta & Greisen 2002)), Sect. 8.4 describes spectral coordinate systems (see Greisen et al. 2006), and Sect. 9 describes the representation of time coordinates (see Rots et al. 2015).

8.2. World coordinate system representations

A variety of keywords have been reserved for computing the coordinate values that are to be associated with any pixel location within an array. The full set is given in Table 22; those in most common usage are defined in detail below for convenience. Coordinate system specifications may appear in HDUs that contain simple images in the primary array or in an image extension. Images may also be stored in a multi-dimensional vector cell of a binary table, or as a tabulated list of pixel locations (and optionally, the pixel value) in a table. In these last two types of image representations, the WCS keywords have a different naming convention which reflects the needs of the tabular data structure and the 8-character limit for keyword lengths, but otherwise follow exactly the same rules for type, usage, and default values. See reference Calabretta & Greisen (2002) for example usage of these keywords. All forms of these reserved keywords *must* be used only as specified in this Standard.

In the case of the binary table vector representation, it is possible that the images contained in a given column of the table have different coordinate transformation values. Table 9 of Calabretta & Greisen (2002) illustrates a technique (commonly known as the "Green Bank Convention[9]") which utilizes additional columns in the table to record the coordinate transformation values that apply to the corresponding image in each row of the table. More information is provided in Appendix L.

The keywords given below constitute a complete set of fundamental attributes for a WCS description. Although their inclusion in an HDU is optional, *FITS* writers *should* include a complete set of keywords when describing a WCS. In the event that some keywords are missing, default values *must* be assumed, as specified below.

- WCSAXES – [integer; default: NAXIS, or larger of WCS indexes *i* or *j*]. Number of axes in the WCS description. This keyword, if present, *must* precede all WCS keywords except NAXIS in the HDU. The value of WCSAXES *may* exceed the number of pixel axes for the HDU.
- CTYPE*i* – [string; indexed; default: '␣' (i.e. a linear, undefined axis)]. Type for the intermediate coordinate axis *i*. Any coordinate type that is not covered by this standard or an officially recognized *FITS* convention *shall* be taken to be linear. All non-linear coordinate system names *must* be expressed in '4–3' form: the first four characters specify the coordinate type, the fifth character is a hyphen ('-'), and the remaining three characters specify an algorithm code for computing the world coordinate value. Coordinate types with names of less than four characters are padded on the right with hyphens, and algorithm codes with less than three characters are padded on the right with blanks[10]. Algorithm codes *should* be three characters.
- CUNIT*i* – [string; indexed; default: '␣' (i.e., undefined)]. Physical units of CRVAL and CDELT for axis *i*. Note that units *should* always be specified (see Sect. 4.3). Units for celestial coordinate systems defined in this Standard *must* be degrees.

[9] Named after a meeting held in Green Bank, West Virginia, USA in 1989 to develop standards for the interchange of single dish radio astronomy data.

[10] Example: 'RA---UV '.

Appendix C: Summary of keywords

This Appendix is not part of the FITS *standard, but is included for convenient reference.*

All of the mandatory and reserved keywords that are defined in the standard, except for the reserved WCS keywords that are discussed separately in Sect. 8, are listed in Tables C.1, C.2, and C.3. An alphabetized list of these keywords and their definitions is available online: `http://heasarc.gsfc.nasa.gov/docs/fcg/standard_dict.html`.

Table C.1: Mandatory *FITS* keywords for the structures described in this document.

Primary HDU	Conforming extension	Image extension	ASCII table extension	Binary table extension	Compressed images[6]	Compressed tables[6]	Random groups records
SIMPLE	XTENSION	XTENSION[1]	XTENSION[2]	XTENSION[3]	ZIMAGE =T	ZTABLE =T	SIMPLE
BITPIX	BITPIX	BITPIX	BITPIX = 8	BITPIX = 8	ZBITPIX	ZNAXIS1	BITPIX
NAXIS	NAXIS	NAXIS	NAXIS = 2	NAXIS = 2	ZNAXIS	ZNAXIS2	NAXIS
NAXISn[4]	NAXISn[4]	NAXISn[4]	NAXIS1	NAXIS1	ZNAXISn	ZPCOUNT	NAXIS1 = 0
END	PCOUNT	PCOUNT = 0	NAXIS2	NAXIS2	ZCMPTYPE	ZFORMn	NAXISn[4]
	GCOUNT	GCOUNT = 1	PCOUNT = 0	PCOUNT		ZCTYPn	GROUPS = T
	END	END	GCOUNT = 1	GCOUNT = 1		ZTILELEN	PCOUNT
			TFIELDS	TFIELDS			GCOUNT
			TFORMn[5]	TFORMn[5]			END
			TBCOLn[5]	END			
			END				

(1)`XTENSION=␣'IMAGE␣␣␣'` for the image extension. (2)`XTENSION=␣'TABLE␣␣␣'` for the ASCII table extension. (3)`XTENSION=␣'BINTABLE'` for the binary table extension. (4)Runs from 1 through the value of `NAXIS`. (5)Runs from 1 through the value of `TFIELDS`. (6)required in addition to the mandatory keywords for binary tables.

Table C.2: Reserved *FITS* keywords for the structures described in this document.

All[1] HDUs		Array[2] HDUs	ASCII table extension	Binary table extension	Compressed images	Compressed tables	Random groups records
DATE	EXTNAME	BSCALE	TSCALn	TSCALn	ZTILEn	FZTILELN	PTYPEn
DATE-OBS	EXTVER	BZERO	TZEROn	TZEROn	ZNAMEi	FZALGOR	PSCALn
ORIGIN	EXTLEVEL	BUNIT	TNULLn	TNULLn	ZVALi	FZALGn	PZEROn
AUTHOR	EQUINOX	BLANK	TTYPEn	TTYPEn	ZMASKCMP		
REFERENC	EPOCH[3]	DATAMAX	TUNITn	TUNITn	ZQUANTIZ		
COMMENT	BLOCKED[3]	DATAMIN	TDISPn	TDISPn	ZDITHER0		
HISTORY	EXTEND[4]		TDMAXn	TDIMn	ZSIMPLE	ZTHEAP	
␣␣␣␣␣␣␣␣	TELESCOP		TDMINn	THEAP	ZEXTEND		
OBJECT	INSTRUME		TLMAXn	TDMAXn	ZBLOCKED		
OBSERVER			TLMINn	TDMINn	ZTENSION		
CONTINUE				TLMAXn	ZPCOUNT		
INHERIT [5]				TLMINn	ZGCOUNT		
CHECKSUM					ZHECKSUM	ZHECKSUM	
DATASUM					ZDATASUM	ZDATASUM	

(1)These keywords are further categorized in Table C.3. (2)Primary HDU, image extension, user-defined HDUs with same array structure. (3)Deprecated. (4)Only permitted in the primary HDU. (5)Only permitted in extension HDUs, immediately following the mandatory keywords.

Table C.3: General reserved *FITS* keywords described in this document.

Production	Bibliographic	Commentary	Observation
DATE	AUTHOR	COMMENT	DATE-OBS
ORIGIN	REFERENC	HISTORY	TELESCOP
BLOCKED[1]		␣␣␣␣␣␣␣␣	INSTRUME
			OBSERVER
			OBJECT
			EQUINOX
			EPOCH[1]

(1)Deprecated.

- The last paragraph of Sect. 4.1.2.3 was corrected to state that the ASCII text characters have hexadecimal values 20 through 7E, not 41 through 7E.

H.3. List of modifications to the latest FITS standard

1. The representation of time coordinates has been incorporated by reference from Rots et al. (2015) and is summarized in Sect. 9. Cross-references have been inserted in pre-existing sections of the Standard (namely in Sect. 4.2.7, 4.3, 4.4.2.1, 4.4.2.2 and 5.4, as well as in various places of Sect. 8, like 8.3 and 8.4.1). New keywords are listed in a rearranged Table 22. Contextually an erratum was applied in Sect. 8.4.1: keywords `OBSGEO-[XYZ]` were incorrectly marked as `OBSGEO-[XYZ]`*a*; the TAI-UTC difference in Table 30 was updated with respect to Rots et al. (2015) taking into account the latest leap second; the possibility of introducing more sources for the solar system ephemerides was re-worded (at the end of Sect.9.2.5 and in Table 31).
2. The continued string keywords described in Sect. 4.2.1.2 were originally introduced as a *FITS* convention since 1994, and registered in 2007. The text of the original convention is reported at `http://fits.gsfc.nasa.gov/registry/continue_keyword.html`. The differences with this standard concern:
   - In the convention, the `LONGSTRN` keyword was used to signal the possible presence of long strings in the HDU. The use of this keyword is no longer required or recommended.
   - Usage of the convention was *not recommended* for reserved or mandatory keywords. Now it is *explicitly forbidden* unless keywords are explicitly declared long-string.
   - To avoid ambiguities in the application of the previous clause, the declaration of string keywords in sections 8, 9 and 10 has been reset from the generic 'character' to 'string'.
   - It is also explicitly clarified there is no limit to the number of continuation records.
   - The description of continued comment field is new.
3. The blank header space convention described in Sect. 4.4.2.4 was used since 1996, and registered in 2014. The text of the original convention is reported at `http://fits.gsfc.nasa.gov/registry/headerspace.html`. It included a *recommendation* about using the convention in a controlled environment, which does not appear in this standard.
4. The `INHERIT` keyword described in Sect. 4.4.2.6 was originally introduced as a *FITS* convention since 1995, and registered in 2007. The text of the original convention is reported at `http://fits.gsfc.nasa.gov/registry/inherit.html`. See also references and practical considerations therein. The differences with the present document concern a more precise RFC-2219 compliant wording in a couple of sentences in Appendix K.
5. The table keywords described in Sect. 7.2.2 and 7.3.2 were originally introduced as a *FITS* convention since 1993, and registered in 2006. The text of the original convention is reported at `http://fits.gsfc.nasa.gov/registry/colminmax.html`. The differences with this standard concern:
   - The exclusion of undefined or IEEE special values when computing maximum and minimum is now *mandatory* while it was *optional*.
   - The original text included the possibility of using the fact `TDMINn` were greater than `TDMAXn` (or `TLMINn` greater than `TLMAXn`) as an indication the values were undefined. This clause has been removed
   - The original text contained usage examples and additional minor explanatory details.
6. The checksum keywords described in Sect. 4.4.2.7 were originally introduced as a *FITS* convention since 1994, and registered in 2007. The text of the original convention is reported at `http://fits.gsfc.nasa.gov/registry/checksum.html`. The differences with this standard concern:
   - The omission of some additional implementation guidelines.
   - The omission of a discussion on alternate algorithms and relevant additional references.
7. The conventions for compressed data described in Sect. 10. were originally introduced as a couple of *FITS* conventions registered in 2007 and 2013. The text of the original conventions is reported at `http://fits.gsfc.nasa.gov/registry/tilecompression.html` for compressed images and at `http://fits.gsfc.nasa.gov/registry/tiletablecompression.html` for compressed binary tables. The differences with this standard concern:
   - In Sect. 10.3.3 the original text for `FZALG`*n* mentioned the possibility that, 'If the column cannot be compressed with the requested algorithm (e.g., if it has an inappropriate data type), then a default compression algorithm will be used instead.' But there is no default algorithm. This is irrelevant for the Standard.
   - In Sect. 10.4 the alias `'RICE_ONE'` is *not* adopted in the Standard as a synonym for `'RICE_1'`.
   - In Sect. 10.4.3 a sentence was left out about requiring additional instructions in PLIO to make it work for more then 2^{12} bits, since we aren't allowing this possibility in the Standard.
   - In Sect. 10.4.4 the reference to a 'smoothing flag' was dropped.
   - Also in Sect. 10.4.4 the *scale factor* is now floating point, while it was originally integer.
   - In Table 36 (and Sect. 10.3.5) the `NOCOMPRESS` algorithm is explicitly mentioned.
8. The Green Bank convention, mentioned in Sect. 8.2 and described in Appendix L, was in use since 1989, and registered in 2010. The text of the registered convention is reported at `http://fits.gsfc.nasa.gov/registry/greenbank/greenbank.pdf` and contains some additional details about the history of the convention.

Appendix K: Header inheritance convention

This Appendix is not part of the FITS *standard, but is included for informational purposes.*

The reserved boolean `INHERIT` keyword described in Sect.4.4.2.6 is optional, but if present it *shall* appear in the extension header immediately after the mandatory keywords. The `INHERIT` keyword *must not* appear in the primary header. Keyword inheritance provides a mechanism to store keywords in the primary HDU, and have them be shared by one or more extensions in the file. This mechanism minimizes duplication (and maintenance) of metadata in multi-extension FITS files.

It *should* only be used in *FITS* files that have a null primary array (e.g., with `NAXIS = 0`). to avoid possible confusion if array-specific keywords (e.g., `BSCALE` and `BZERO`) were to be inherited.

When an application reads an extension header with `INHERIT = T`, it should merge the keywords in the current extension with the primary header keywords. The exact merging mechanism is left up to the application. The mandatory primary array keywords (e.g., `BITPIX`, `NAXIS`, and `NAXISn`) and any `COMMENT`, `HISTORY`, and blank keywords in the primary header *must not* be inherited. It is assumed also that the table-specific keywords described in Sect. 7.2 and 7.3, and the table-specific WCS keywords described in Sect. 8, cannot be inherited since they will never appear in the primary header. If the same keyword is present in both the primary header and the extension header, the value in the extension header *shall* take precedence. If `INHERIT = F` in an extension header, the keywords from the primary header *should* not be inherited.

An application which merely reads a FITS file is authorized by `INHERIT = T` to look up in the primary HDU for an expected keyword not found in the current HDU. However if the application writes out a modified file, it has to be very careful to avoid unwanted duplication of keywords, and preserve the separation of primary and extension headers, namely if an application modifies the value of an inherited keyword while processing an extension HDU, then it is recommended to write the modified value of that keyword into the extension header, leaving the value of the keyword in the primary header unchanged. The primary array keywords should only be modified when the intent is to explicitly change the value that will subsequently be inherited in the extensions.

Also if the FITS file is read in sequentially (e.g., from tape or Internet downloads), the reader would need to cache the primary header in case it turns out that a later extension in the file uses the INHERIT keyword.

Appendix L: Green Bank convention

This Appendix is not part of the FITS *standard, but is included for informational purposes.*

The Green Bank convention was developed at a meeting in October 1989 at Green Bank Observatory, West Virginia, to discuss the use of FITS for single dish radio astronomy data, and has since been widely used in conjunction with the SDFITS convention. It was devised primarily to record WCS keywords independently for each row of a table containing an image array column, but subsequently it has found more general application.

The basic idea is that of expanding header keywords into binary table columns, and vice versa, of collapsing unvarying binary table columns into header keywords.

For example, the standard header keyword `DATE-OBS`, which records the date and time of observation, could be expanded into a column with `TTYPEn = 'DATE-OBS'` to record the date and time independently for each row of a binary table. Conversely, a binary table column with `TTYPEn = 'HUMIDITY'` containing the same value in each row, could be collapsed into a keyword, `HUMIDITY`, that recorded the constant value.

When the Green Bank convention is used (and arguably otherwise) a keyword should not coexist with a column of the same name within a single binary table. Should this situation occur, the column value takes precedence over the keyword.

When expanding keywords into columns, the Green Bank convention applies to all FITS keywords that may appear in a binary table except for the following, most of which describe the structure or identity of a binary table HDU: `XTENSION`, `BITPIX`, `NAXIS`, `NAXISn`, `PCOUNT`, `GCOUNT`, `TFIELDS`, `EXTNAME`, `EXTVER`, `EXTLEVEL`, `TTYPEn`, `TFORMn`, `TUNITn`, `TSCALn`, `TZEROn`, `TNULLn`, `TDISPn`, `THEAP`, `TDIMn`, `DATE`, `ORIGIN`, `COMMENT`, `HISTORY`, `CONTINUE`, and `END`.

In order to collapse a column into a keyword, the name of the column (given by `TTYPEn`) must be a valid keyword name, and the column's constant value must be amenable to representation as a valid keyvalue.

Software that implements the Green Bank convention must take into account the possibility that any "keyword" (apart from those on the proscribed list), such as `DATE-OBS`, may change value from row to row of a table. Moreover, when searching the header for a particular keyword, it must first consider the values of the `TTYPEn` keywords in case the desired keyword has been expanded as a column. Likewise, it must consider each header keyword potentially as a collapsed column, so that a request for the value in a particular row or rows of the nonexistent `'HUMIDITY'` column would be satisfied by the value of the `HUMIDITY` keyword.

References

Note: Many of these *FITS* references are available electronically from the NASA Astrophysics Data System (ADS) and/or the *FITS* Support Office web sites at
`http://adswww.harvard.edu` and
`http://fits.gsfc.nasa.gov/fits_documentation.html`.

Allen, S. & Wells, D. 2005, IETF RFC 4047, `http://www.ietf.org/rfc/rfc4047.txt`

ANSI 1977, *American National Standard for Information Processing: Code for Information Interchange*, ANSI X3.4–1977 (ISO 646) New York: American National Standards Institute, Inc.

Braden, R. T., Borman, D.A., and Partridge, C. 1988 ACM Computer Communication Review, 19, no. 2, 86, IETF RFC 1071, `https://tools.ietf.org/html/rfc1071`

Bradner, S. 1997, IETF RFC 2119, `http://www.ietf.org/rfc/rfc2119.txt`

Bunclark, P. & Rots, A. 1997, *Precise re-definition of* `DATE-OBS` *Keyword encompassing the millennium*, `http://fits.gsfc.nasa.gov/year2000.html`

Calabretta, M. R. & Greisen, E. W. 2002, A&A, 395, 1077

Calabretta, M. R. & Roukema, B. F. 2007, MNRAS, 381, 865

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

Cotton, W. D., et al. 1990, *Going AIPS: A Programmer's Guide to the NRAO Astronomical Image Processing System*, Charlottesville: NRAO

Deutsch P. 1996, RFC 1951, Network Working Group; available online: `http://tools.ietf.org/html/rfc1951`

Folkner, W. M., Williams, J. G., & Boggs, D. H. 2009, Interplanetary Network Progress Report 42-178, available online: `http://tmo.jpl.nasa.gov/progress_report/42-178/178C.pdf`