

Starting on February 19, 2021, the format of the headers of the exported FITS files produced changed. Before this date, the files produced were not FITS-compliant. After this date, the files produced were compliant with version 4.0 of the FITS Standard ([https://fits.gsfc.nasa.gov/standard40/fits\\_standard40aa-le.pdf](https://fits.gsfc.nasa.gov/standard40/fits_standard40aa-le.pdf)). In addition to changes needed to achieve compliance, several other changes were made to enhance readability (in preparation for final archiving).

A description of the changes made follows.

#### FITS Keyword Changes

-----

Many of the following changes were made so that the exported FITS files are now FITS-standard compliant. Some changes were made to enhance readability of the FITS header, or to retain data-series information that would otherwise be lost.

#### + keyword value comments

- previously, keyword values lacked comments; example:

```
RSUN_OBS= 967.630554
```

- currently, a comment is included, with the following syntax (square brackets denote optional components):

a '/' character occupies the byte after the byte occupied by the last value character

a ' ' character occupies the byte after the byte occupied by the '/' character

[ a string describing the physical units, surrounded by square brackets, occupies the bytes after the byte occupied by the last ' ' character; a ' ' occupies the byte after the byte that the ']' occupies ] (if the value has a physical unit)

a comment occupies the bytes after the byte that the last ' ' occupies

[ a ' ' occupies the byte after the byte that the last comment character occupies; a string containing the DRMS keyword name in curly braces occupies the bytes after the byte that the last ' ' occupies ] (if the FITS keyword name does not equal the DRMS keyword name - to retain the DRMS the mapping from FITS keyword name to DRMS keyword name)

- current examples:

```
RSUN_OBS= 945.91434453710849 / [arcsec] Apparent radius of the Sun seen by SDO  
DATE-OBS= '2016-05-09T10:22:43.900' / [ISO] Observation date {DATE__OBS}
```

#### + number keyword values

- previous, number keyword values were not right-justified in bytes 11 through 30 (justification is a FITS requirement); example:

```
RSUN_OBS= 967.630554
```

- currently, number values are right-justified; example:

```
RSUN_OBS= 945.91434453710849 / [arcsec] Apparent radius of the Sun seen by SDO
```

+ missing keyword values

- previously, NaN keyword values, the value to indicate a missing floating-point value in DRMS, were represented by a non-quoted value of three characters: nan; this was not FITS-compliant; example:

```
OSCMEAN= nan
```

- previously, non-floating point keyword values were represented by compliant numbers, like -2147483648 for a missing integer value; example:

```
ROI_LLY1=          -2147483648
```

- currently, a consistent, FITS-compliant method is used to denote all missing values, regardless of data type:

a FITS keyword name (1 to 8 chars in length only) occupies the first 8 bytes

an '=' character occupies byte 9

a ' ' (a space) character occupies byte 10

' ' characters occupy the next several bytes (determined by FITSIO keyword-writing routines)

a '/' character occupies the byte after the byte occupied by the last ' ' character

a ' ' character occupies the byte after the byte occupied by the '/' character

the string "(MISSING)" occupies the next 10 bytes after the byte occupied by the '/' character

a ' ' character occupies the byte after the byte that the last ')' character occupies

a comment occupies the bytes after the byte that the ' ' occupies

[ a ' ' occupies the byte after the byte that the last comment character occupies; a string containing the DRMS keyword name in curly braces occupies the bytes after the byte that the last ' ' occupies ] (if the FITS keyword name does not equal the DRMS keyword name - to retain the DRMS the mapping from FITS keyword name to DRMS keyword name)

- current example:

```
CALVER32=          / (MISSING) Calibration Version {ROI_NWIN}
```

+ date keywords

- previously, FITS-reserved date keyword values (DATE, DATE-OBS, etc.) were not consistent (various representations were used: ISO 8601, TAI, with/without time zones, depending on data series); the Standard requires that the value of these keyword be have the form YYYY-MM-DDThh:mm:ss[.sss...] WITHOUT a time zone specified; the value is interpreted as a UTC time; it should not be interpreted as an ISO time (which would imply the string was local time)

- currently, dates are all represented by UTC times of the form YYYY-MM-DDThh:mm:ss[.sss...]; examples:

```
DATE      = '2017-11-03T13:31:59.000' / [ISO] HDU creation date
```

```
DATE-OBS= '2016-05-09T10:22:43.900' / [ISO] Observation date {DATE__OBS}
```

+ keyword exponent-notation values

- previously, floating-point exponents were denoted by lower-case 'e' and 'd' characters, which are not FITS-compliant
- currently, upper-case 'E' and 'D' are used

+ long keyword values

- previously the length of certain string keywords, like `SOURCE`, exceeded the number of bytes available so their values were truncated; example:

```
SOURCE = 'hmi.lev1[:#195382928,#195382880,#195382832,#195382784,#195382736,#19'
```

- currently, the FITS long-string convention is used (the string continues to the next line with one or more CONTINUE keywords; each line, other than the last one, ends in an '&'); example:

```
SOURCE = 'hmi.lev1[:#195382928,#195382880,#195382832,#195382784,#195382736,#1&'
CONTINUE '95382688,#195382640,#195382592,#195382964,#195383024,#195383072,#19&'
CONTINUE '5383120,#195383168,#195383216,#195382929,#195382881,#195382833,#195&'
...
CONTINUE '3199,#195383247]' / level 1 filtergrams used to produce the observa
```

+ precision of floating-point keyword values

- previously, a user-defined format string was used to generate the values; example:

```
DATAMEAN= -1473.85
```

- currently, the "maximum" precision used the FITSIO writing routines is used; example:

```
DATAMEAN= -1473.8548583984375 / [m/s] Mean value from pixels within 99% of sola
```

- a separate document will be provided to describe actual precision

+ DRMS ID

- previously, no keyword value uniquely identified each and every data file
- currently, such a keyword, DRMS\_ID, uniquely identifies each and every segment file; its value is generated by concatenating the DRMS series, the series recnum, and the segment, with colons separating the components; example:

```
DRMS_ID = 'hmi.V_720s:485458:Dopplergram' / DRMS ID
```