See change log on last page(s).
Items suggested for change or questions are highlighted in colors.

JSOC Keywords used for metadata

Introductory Remarks

Keyword Syntax Standards

The DRMS system is based on data organized as records containing keyword tagged metadata and data arrays stored in named “segments”. To be useful the set of commonly used keywords must be agreed to by the majority of users and should be easily recognized by researchers familiar with SOHO/MDI and IDL SolarSoft (SSW) data formats. Both MDI and SSW are based on the FITS standard for external file formats.

Naming Syntax Issues

The space of names for keywords is constrained by implementation details with various heritages. In some language bindings the internal name seen by the programmer differs from the names actually stored in the data files. (E.g. the FITS “DATE-OBS” keyword is “DATE-OBS” in a file but is referenced as “DATE_d$OBS” in SolarSoft IDL code.) Similarly DRMS supports a limited set of characters to be used in keyword names since those names are also column names in PostgreSQL database tables. The FITS protocol, a standard for data exchange, has limits on characters (upper case letters, digits, minus-sign “-“, and underscore “_”) and word length (8 characters, blank filled) starting left adjusted on a line which are based on the 80 column punched card heritage. Some language bindings for FITS remove trailing blanks in another example of internal and external representation differences. DRMS limits keywords to 254 characters in length but also limits them to ASCII letters, digits, and the underscore ‘_’ character. In DRMS case is preserved but ignored. Names must begin with a letter.

Since DRMS names are not allowed to have a “-“, and some of the FITS Standard keywords do require a “-“, it is clear that the internal and external names for at least some keywords must differ. (This is similar to the IDL problem with “-“). Additionally some commonly used lists of FITS keywords are inconsistent with each other in the choice of keyword names for the same quantity and some even are inconsistent for the meaning of keywords with the same name. Some of the inconsistencies are due to poorly specified usage guidelines. An example is again “DATE-OBS” where MDI uses it to identify the time at which the image represents the measured quantity and SSW uses it to represent the start of the interval over which the observation was gathered. However, for spatial information MDI, SSW, FITS, and now JSOC use keywords which reference the center of the tagged location, i.e. the center of a pixel vs. the lower left corner of a pixel recognizing that a pixel has a finite extent in image space. Given these issues we must be careful to define the working set of internal DRMS keywords and the required mapping to FITS keywords for export to SSW and other “consumers”. Export to MDI should not be an issue since we will eventually port MDI analysis over to DRMS. But we expect to export data as FITS files to users of GONG based programs as well as SSW, so we need to be careful.

DRMS and FITS

In addition to keyword issues, there are other concerns that must be addressed when exporting DRMS data into FITS files. The basic DRMS record structure is certainly a subset of the rich FITS format but there are some limitations in addition to mapping 254 char keyword names to 8 chars. Primarily the issues concern links which are supported by DRMS. A DRMS record may contain zero or more links to other DRMS records. Furthermore, keywords and segments in the linked records may be referenced from the record with the link as if they were in that record rather than in the link target record. On export keyword links will be followed but the record links themselves will vanish. Links may be static or dynamic. A static link points to a specific unchanging DRMS record. A dynamic link points to DRMS record with a particular prime index value. If the link target record is updated, the dynamic link will point to the most recent version. Upon export, dynamically linked keywords get the values valid at export time. A DRMS record may contain multiple data arrays with varying dimension. If we restrict our use of FITS exports to Simple FITS files we will need to decide how to handle multiple array records. Some DRMS records will contain no array data but will have a record per time step. Such dataseries could reasonably be exported as FITS tables. We will need to decide if this will be implemented. The MDI FITS reader does not support FITS tables. If FITS tables are commonly used in SSW code we should be able to support them as an export product. Multiple images/arrays per record can also be handled in FITS as extensions. As a minimum, we should be able to export SSW compliant simple images as single FITS files. We need some decisions on these topics. One simple thing we can add is to add a keyword per link with the contents a string giving the linked record query.

Image Coordinate Mapping Keywords

We use FITS standard keywords to describe the mapping between pixel-space and physical space, the so called “world coordinate system” or WCS. (See “Coordinate systems for solar image data”, W.T. Thompson, A&A 449, 791-803, 2006 for definitions from an observers perspective) In WCS each physical coordinate axis is described by a set of keywords specifying the type of coordinate, physical units, and mapping onto array elements assumed to be image pixels.

There is a design weakness in the FITS paper-1 coordinate mapping rules for mapping telescope images into locations in the sky and then onto the Sun. The essence of the issue has to do with an assumed known mapping between pixel addresses and arc-seconds. The FITS WCS Paper (Greisen and Calabretta, A&A 395, 1061-1075, 2002) definitions of CRPIXj, CRVALi, and CTYPEi combined with one of CROTAj and CDELTi; PCi_js and CDELTi; or just CDi_js imply that the mapping to arc-seconds is well known. (These keywords are described below). This is seldom the case. For some instruments and analyses it may be that an estimate of this relationship is sufficient. For helioseismology is it not sufficient. *A priori* estimates of the plate scale are also not sufficient for comparing pixel-sized solar features between instruments. The correct mapping between a pixel address and a location in the Sun’s atmosphere is very complex even for in instrument where the light originates from a volume with thickness small compared to a pixel’s horizontal scale. For EUV imaging of optically thin regions it has an additional uncertainty of height. What one would ideally want is a mapping of pixel brightness to a solar angular coordinate (e.g. latitude and longitude in some well defined coordinate system) and distance from the Sun’s center, or height above some well defined surface.

However, we will adopt the standard procedures and adopt a set of keywords common to both HMI and AIA. The computation of the basic coordinate mapping keywords must be different because of both need and practicality. While HMI both needs an accurate plate scale and can measure it, AIA neither needs nor can easily measure it to the accuracy needed by HMI.

Basic HMI and AIA Level-0 Image Keywords Approach

We propose the following scenario:

AIA: *a FIXED value for the telescope plate scale* for each telescope is assumed. This value will be stored in the keyword IM_SCALE. At the first opportunity (level-1.0) CTYPE will be set to HPLN-TAN and HPLT-TAN, with CROTA2 to be logically applied *before* assigning the implied labels of X and Y used to map array index 1 and 2. The HPLT or "SOLARY" direction will be the projection of the Carrington solar rotation axis onto the plane of the sky (+ is north) and HPLN or "SOLARX" is perpendicular to that also in the plane of the sky, (+ west on Sun which is roughly in the direction of Earth orbit motion). Once this is done XCEN and YCEN can be computed.

HMI: *a FIXED value for the radius of the Sun* in meters combined with a measured average radius (pixels) of the solar image using a non-changing definition of the solar limb, combined with the known distance between the telescope and the solar center (not photosphere). Here the keywords "R_SUN", "CRPIX1", and "CRPIX2" will contain the key information from which the other values are computed. R_SUN, CRPIXj are all in pixels with center of the lower left pixel of the array set to 1.0, 1.0. The older MDI X0 and Y0 are the location of the solar disk center in the image and are X0 = CRPIX1-1 and Y0 = CRPIX2 -1. R_SUN is first determined in level1.0 processing so these keywords are absent prior to level1.0 data. They are included in the level-1 data only for reporting the measured quantities. They are not propagated to higher processing levels.

Then in order, information that is available prior to level-1.0 processing:

From SDO attitude and orbit information:

SAT_ROT = SDO roll such that SAT_ROT is degrees of rotation of the image of the Sun's pole projection onto the CCD with Sun's N pole CW from the "Y" CCD axis for positive SAT_ROT. I.e. SAT_ROT is positive for a CCW roll of SDO when viewed from behind SDO looking toward the Sun. This will be determined from SDO attitude data.
DSUN_OBS distance to Sun center from spacecraft in m, c. 1.5E11.
RSUN_REF = radius of Sun in m, agreed upon standard, c. 6.96E8 but must be consistent with WAVELNTH keyword.

For each CCD camera:

INST_ROT = Telescope roll angle is the angle between the instrument CCD Y axis and the SDO Z axis. This is a calibration (nearly constant) determined independently for each of the AIA and HMI cameras. The sign convention should be the same as for SAT_ROT after any required image flipping to allow solar west to be to the right when solar north is up.

Now, only at level-1.0 we can provide the following:

For AIA:

X0, Y0 are the array addresses of the center of the solar disk image and are computed from commanded pointing and known offsets, pixel address of the science reference boresite with telescope specific corrections.
IM_SCALE is the predefined AIA plate scale in arc-seconds per pixel.
R_SUN is computed from DSUN_OBS, RSUN_REF, and IM_SCALE.

For HMI:

X0, Y0 are computed from a fit to image.
R_SUN is computed from the same fit to each image.
IM_SCALE is computed from R_SUN and DSUN_OBS and RSUN_REF.

For both AIA and HMI for lev1.0 and above:

CDELT1 = **CDELT2** = IM_SCALE if full resolution, else scaled from IM_SCALE. IM_SCALE remains the plate scale of the original image while CDELT1 and CDELT2 are set to be correct for a rescaled image.
CROTA2 = SAT_ROT + INST_ROT.
CRPIX1, **CRPIX2** set according to processing while X0, Y0 remain unchanged to indicate the position of the raw image on the CCD. Note that CRPIXj start at 1.0 while X0,Y0 start at 0.0. Thus CRPIX1 = X0 + 1, and CRPIX2 = Y0 + 1.
CRVAL1 = **CRVAL2** = 0.0, 0.0
XCEN, YCEN computed from above.

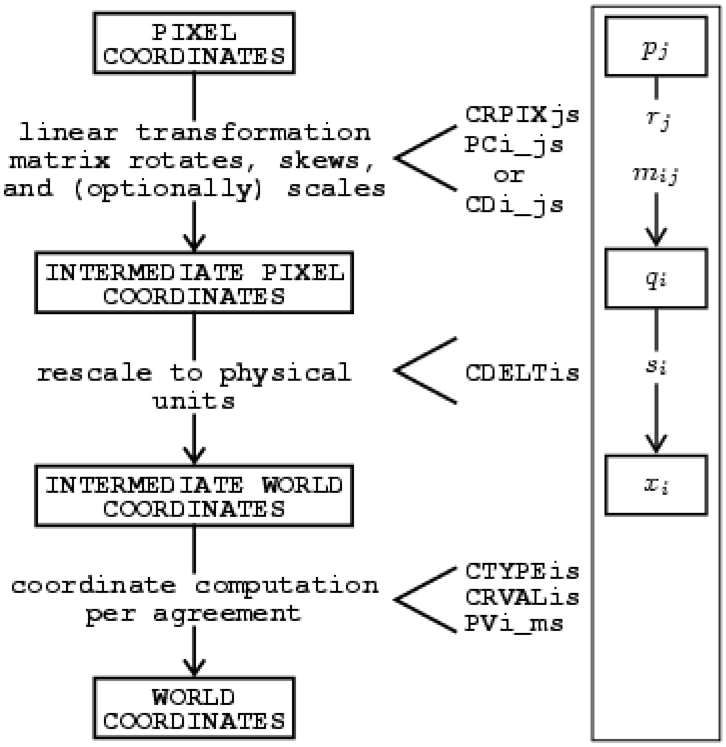

PIXEL COORDINATES
linear transformation matrix rotates, skews, and (optionally) scales
CRPIXjs
PCi_js
or
CDi_js
INTERMEDIATE PIXEL COORDINATES
rescale to physical units
CDELTis
INTERMEDIATE WORLD COORDINATES
coordinate computation per agreement
CTYPEis
CRVALis
PVi_ms
WORLD COORDINATES
p_j
r_j
m_{ij}
q_i
s_i
x_i

Figure 1: FITS WCS Coordinate Mapping Sequence from WSC paper. Shows the relations between the world coordinates indexed by *i* and the array coordinates indexed by *j*.

Keyword Recommendations for JSOC Data

While some keywords are expected in all or almost all data products, most are only included at a certain level of processing and are not propagated beyond some other level of processing. In the descriptions below, keywords are grouped in sets that should usually occur or not occur together.

See the table below to determine which sets are expected to be present for which products.

	TLM files	Lev-0	Lev-1	HMI lev1.5	AIA lev1.5
1. FITS basic		X	X	X	X
2. FITS exports			X	X	X
3. Data overview		X	X	X	X
4. WCS			X	X	X
5. Statistics		X	X	X	X
6. Image position			X		
7. SSW special			?		?
8. Observer location			X	X	X
9. JSOC tlm	X				
10. JSOC Lev0		X			
11. JSOC Lev1			X		
12. HMI Observables				X	
13. AIA Lev1.5					X

1. **FITS required keywords**

   These describe the external representation of the data and do not appear as DRMS keywords. The information is used for the basic structure of the stored data and is handled via other constructs in DRMS. These keywords will be generated as part of the export process into FITS files.

   a. **SIMPLE** Boolean, always T
   b. **BITPIX** integer, 16, 32, -32, or -64
   c. **NAXIS** integer
   d. **NAXISn** integers, n is 1 to NAXIS
   e. **END** no value part

   The following 3 keywords are only made when BITPIX > 0.

   f. **BSCALE** float, Phys_val = BZERO + BSCALE * Stored_val.
   g. **BZERO** float
   h. **BLANK** integer

2. **Exported FITS files special keywords**

   a. **RECORD** text DRMS record specification for the data, this is required only in exported FITS files, format uses "<seriesname>[<primekey>]..{segname}" notation. Same text as 'show_info –i …'
   b. **L_XXXXXX** Link query, XXXXXX built from link name using standard export rules. Providing access to linked records allows traceback when needed.

3. Reserved keywords for source of data

These will be generated for HMI and AIA data for all records. Note on times: all times are stored internally in DRMS as type TIME which is a double. They are printed by default in the format indicated in the JSOC Series Definition (jsd) file. The formats below are the recommended jsd specs. Most but not all of these are FITS reserved keywords.

a. **DATE** text, Date and time when the file is created. Must use specific format. This format is yyyy.mm.ddThh:mm:ss[.sss] in ISO-8601 format UTC. Stored as double TAI time in DRMS.

b. **DATE__OBS** text, Date and time when observation of this image was started. Uses DATE format. DATE-OBS will be computed as T_OBS – EXPTIME/2.0. Stored as a double in DRMS. NOTE: FITS keyword will be **DATE-OBS** upon export. **NOTE**: *DATE_OBS, the old SOHO keyword is* ***not present, needed ,nor used for SDO****. Differs from DATE-OBS by less than 0.1 second for the SDO orbit. Was c. 5 seconds for SOHO.*

c. **TELESCOP** text, Source telescope for data. In the normal FITS hierarchy this is more encompassing than an instrument, which is a part of a telescope. We will use either: “SDO/HMI” or “SDO/AIA” as values.

d. **INSTRUME** text, Instrument (within the telescope in normal FITS usage). Suggestion is to use this field for the AIA telescope/camera combination and for the HMI camera identification. Thus the values will be: ATA_1, ATA_2, ATA_3, ATA_4, HMI_SIDE1, HMI_FRONT2, or HMI_COMBINED. INSTRUME contains the same information as the integer value in the CAMERA keyword.

e. **WAVELNTH** int, Wavelength of observation in Ångstroms!. This keyword will be a constant, 6173 for HMI and for AIA one of 335, 131 for camera 1; 193, 211 for camera 2; 171, 1600, 1700, 4500 for camera 3; 94, 304 for camera 4. It will NOT be set to a higher accuracy for each HMI filtergram. The expected use is for e.g. VSO searches to find Fe-I 6173 data, not to find a particular filtergram. We have chosen to use the non-ISO units of Ångstroms here simply to make it easier to use them for database searches on integers instead of floats.

f. **CAMERA** integer describing the camera used, AIA: 1-4, HMI: 1-2. For HMI the value 3 is used for quantities computed by combining images from both cameras.

g. **BUNIT** text, physical units of data. Level0 CCD image data will use “DN”, higher level data will use e.g. “m/s”.

h. **ORIGIN** text, Location where file was made. e.g. “SDO/JSOC-SDP”.

i. **CONTENT** text, a description of the content of the record/file. For level0 data this could be “filtergram”. For higher level could be e.g. “Dopplergram” or “Synchronic Frame”, or “Carrington Synoptic Map” etc. Will often be information in JSOC data series descriptive note.

j. **HISTORY** text, processing history of data. In DRMS this will be a single keyword which may contain embedded newline chars. Upon export it will be converted to as many HISTORY cards as needed.

k. **COMMENT** text, commentary on the data. In DRMS this is a single keyword which may expand to multiple cards when exported as FITS files.

l. **QUALITY** integer, Bit flags for various kinds of badness. If top bit is set, '0X8000000', then no image is present. If there will never be an image for a slot in a 'sloted' time series, the value '0XC0000000' is used.

m. **BLD_VERS** int, Code release build number of program that created this record.

n. **SOURCE** text, DRMS query that points to the source record, optional and most useful when there is a one-to-one mapping.

4. Image statistics

a. **DATAVALS** integer, number of non-missing data values present. Note DATAVALS + MISSVALS may be less than the product of the array dimensions for lower level products since array elements not present in the telemetry stream from the instruments are not counted as either DATAVALS or MISSVALS.

b. **MISSVALS** integer, number of array elements containing NaN or BLANK values that could be expected to be present in “perfect” data. I.e. missing packets will generate MISSVALS but “corners” of HMI images will not.

c. **TOTVALS** integer, sum of DATAVALS + MISSVALS. Not expected for AIA data.

d. **NERRORS** int, "Number of decompression errors"

e. **NPACKETS** int, "Number of packets in image"

f. **DATAMIN** double, "Minimum value from all pixels"

g. **DATAMAX** double, "Maximum value from all pixels"

h. **DATAMEDN** double, "Median value from all pixels"

i. **DATAMEAN** double, "Mean value for all pixels"

j. **DATARMS** double, "Rms deviation from the mean value of all pixels"

k. **DATASKEW** double, "Skewness from the mean value of all pixels "

l. **DATAKURT** double, "Kurtosis of all pixels"

5. Image coordinate mapping keywords, FITS WCS standard

Specifies mapping from array axes (***j***) to image axes (***i***). See Appendix A.1 for more information.

a. **CTYPE*i*** text, specifies type of image axis ***i*** for other Cxxxx keywords.
b. **CRPIX*j*** float, Reference pixel for array axis ***j***. First pixel is number 1 (not 0).
c. **CRVAL*i*** float, Physical value for image axis ***i*** at the center of the pixel.
d. **CDELT*i*** float, Physical increment per index value for image axis ***i***
e. **CUNIT*i*** text, Physical units for positions on image axis ***i***.
f. **CROTA2** float, Rotation needed for array axes to get to image axes. Unit is degrees. Since for a simple rotation, CROTA is specified for the "latitude" axis we take it to be the j==2 so only CROTA2 will be specified. CROTA2 is defined such that it is positive when the camera is rotated CCW with respect to a fixed Sun, or the Sun's pole is rotated clockwise WRT the frame. Note this is opposite to the normal astronomical convention where a positive position angle for the Sun in the sky corresponds to a CCW rotation of the Pole WRT North in the Sky. (i.e. Dec 21 has P=+7, near Dec7 but with the Earth tilt part of P at zero). For lev0 and lev1 **CROTA2** will be the sum **SAT_ROT** + **INST_ROT**. With this definition **CROTA2** is the negative of MDI **SOLAR_P**. For higher levels which are often rotated to have solar north up, or in some other desired orientation CROTA2 will be appropriate for those products.
g. **CRDER*i*** float, Estimate of random error in coordinate *i* expressed in CUNIT*i*.
h. **CSYSER*i*** float, Estimate of systematic error in coordinate *i* expressed in CUNIT*i*.
i. **PV2_1** float, Exists for Cylindrical Equal Area maps, i.e. longitude vs sine latitude Carrington maps. Contains λ which is $\cos^2$(conformal_latitude).
j. **WCSNAME** string, WCS system name.

These Cxxxxx keywords may have multiple sets present. If so the sets beyond the first have a single letter suffix (denoted 'a') indicating the set. Additionally a WCSNAMEa keyword should be added to identify the set. E.g. if a single second set is present then the additional keywords will be: WCSNAMEA, CTYPE1A, CTRYP2A, CRPIX1A, CRPIX2A, CRVAL1A, CRVAL2A, etc.

6. Image Location Keywords

These keywords are generated in Level-1 processing and are used to compute the initial set of WCS coordinate keywords. These are not expected to be used after level-1 but are present to document the as-observed and measured image location, spacecraft roll, etc.

a. **SAT_ROT** float, degrees, position angle of solar pole wrt the SDO Z axis. This value is required as input to the CROTA calculation. SAT_ROT is positive for a CCW roll of SDO when viewed from behind SDO looking toward the Sun.

b. **INST_ROT** float, degrees, rotation of the camera Y axis from the SDO Z axis. INST_ROT is positive for a CCW roll of SDO when viewed from behind SDO looking toward the Sun.

c. **X0** float, x-axis location of solar disk center in pixels, start 0.0.

d. **Y0** float, x-axis location of solar disk center in pixels, start 0.0. For HMI X0 and Y0 are added at level-1.0 and combined with R_SUN are the source information to compute the values for the coordinate information from this level and higher. X0 and Y0 are wrt the center of the lower left pixel. NOTE that X0 and Y0 are the locations of Sun center in the image array and are NOT related to EW or NS positions on the Sun. They are then the point about which CROTA2 can be applied such that a rotated image will have valid CRVAL, XCEN, and YCEN keywords. X0 and Y0 contain the same information as CRPIX1 and CRPIX2 ONLY for the level-1.0 unmapped images. After any registration, scale changes, mapping, etc., the CRPIXj keywords will reflect the new image while the X0,Y0, R_SUN, IM_SCALE, INST_ROT, and SAT_ROT will, when present, continue to describe the original CCD image.

e. **IM_SCALE** float, arc-sec per CCD pixel value for the particular instrument. This value will be used for the estimate of CDELT for AIA and will be computed from R_SUN and DSUN_OBS for HMI. Note that IM_SCALE does not change when the image is rebinned on the ground. It is a characteristic of the data as observed.

f. **R_SUN** float, Radius of the Sun’s image in pixels.

7. SSW image location keywords

These keywords are expected by SolarSoft IDL code. They are redundant with other keywords but may be included the DRMS list to support direct SolarSoft usage until SSW code is upgraded to look for WCS keywords first.

a. **XCEN** Location on the Sun of the center of the field of view in the array in CDELT1 units (usually arc-sec). Only makes sense for reduction levels where CTYPE1 is e.g. "HPLN-TAN". Is computed from a = CROTA2; XCEN = CRVAL1 + CDELT1*cos(a)*((NAXIS1+1)/2 - CRPIX1) -CDELT2*sin(a)*((NAXIS2+1)/2 - CRPIX2)

b. **YCEN** Is computed from a = CROTA2; YCEN = CRVAL2 + CDELT1*sin(a)*((NAXIS1+1)/2 - CRPIX1) + CDELT2*cos(a)*((NAXIS2+1)/2 - CRPIX2)

8. **Observer Location**
   a. **DSUN_OBS** float, Distance from Sun's center to SDO in m.
   b. **DSUN_REF** float, AU if used in calculations, usually absent but will be 149,597,870,691.0m.
   c. **RSUN_REF** float, Radius of Sun in m, depends on wavelength. For HMI we will adopt 696,000,000m.
   d. **RSUN_OBS** float, angular radius of Sun in arc-sec. Corresponds to arcsin(RSUN_REF/DSUN_OBS).
   e. **CRLN_OBS** float, Carrington longitude of observer, same as SOI **OBS_L0.**
   f. **CRLT_OBS** float, Carrington latitude of observer, same as MDI **OBS_B0**.
   g. **HGLN_OBS** float, "Stonyhurst" longitude from CM of the observer. + to west. This number can be added to the Carrington longitude of disk center (from the Earth's perspective) modulo 360 to get CRLN_OBS. This number is nearly zero for SDO observations (equiv to 170km at photosphere). HGLN_OBS is zero when the observer is on the Sun-Earth line. (Thompson, 2006, is ambiguous about the time at which CRLN_OBS is computed compared to the time at which HGLN_OBS is computed, but SDO is only 0.12 seconds from Earth center in which time the Sun rotates about 240m at the equator).
   h. **CAR_ROT** integer, Carrington rotation number of CRLN_OBS. This is the same as MDI **OBS_CR**.
   i. **OBS_VR** float, velocity of observer in radial direction, + is away from Sun.
   j. **OBS_VW** float, velocity of observer solar-westward, + in rough direction of Earth orbit.
   k. **OBS_VN** float, velocity of observer solar-northward (heliographic), + to north.

9. JSOC telemetry file common keywords

a. **T_PACKET** time (double) version of telemetry packet time in TAI. Converted to text and shown as UT on export or when printed. For images, T_PACKET is the time in the first packet of the image. T_PACKET is slotted on some short step for housekeeping packets.

10. JSOC Lev0 common keywords

a. **FSN** integer containing "Filtergram Sequence Number". For each instrument HMI or AIA the FSN is a unique number for each original image produced by the instrument. FSN will be used as the primary index for level-0 data. For HMI the FSN keyword does not go past level1.0.

b. **EXPTIME** float, seconds, Exposure duration. Often T_OBS = DATE-OBS + EXPTIME/2.0. Since this is often used in combination with DATE-OBS to get "T_OBS" we will define it to be so. Therefore, for single filtergrams EXPTIME will be the actual shutter open time but for products derived from the observation cadence to represent the duration over which the observable samples. For level 1.5 and above it is the time span represented by the observable. I.e. for Dopplergrams measured in 45-seconds or with a 45-second cadence, both EXPTIME and CADENCE may be 45s. If a time series is compact EXPTIME will often be the same as e.g. T_REC_step. DATE-OBS will be computed from T_OBS and EXPTIME.

c. **T_OBS** time (double) version of time of observation in TAI internal to DRMS. On export into FITS headers this is converted to a string in UT from the internal TAI and represents the center of the time span over which the observation is made or is representative. External version is text in JSOC (sprint_ut) format "YYYY.MM.DD_hh:mm:ss{.sss}_UTC" vs ISO-8601 format. T_OBS is later than DATE-OBS by half the "integration" time for level-1 and higher but may be an approximation for level-0 data. (e.g. may be the first packet time). T_OBS will be the primary index for Level-1 time series that are not appropriately described as a slotted-time series where T_REC is the preferred primary index.

d. **TLMDSNAM**string, Telemetry series name containing first packet

e. **IMGAPID** int, Image "APplication ID"

f. **BITSELID** int, Bit Selecte ID, from AHBITID or HHBITID

g. **COMPID** int, Compression ID

h. **TAPCODE** int, 'Take a picture' code

i. **HMI specific lev0** Many special HMI lev0 keywords are extracted from the telemetry header and the Image Status Packet. These are detailed at http://XXXX

j. **AIA specific lev0** Many special AIA lev0 keywords are extracted from the telemetry header and the Image Status Packet. These are detailed at http://XXXX

11. JSOC Lev1 common keywords

a. **T_OBS** From Lev0
b. **FSN** From Lev0:
c. **EXPTIME** From Lev0
d. **QUALLEV0** int, "Quality word" copied from lev0

12. HMI Observables keywords, aka lev1.5

a. **T_OBS**
b. **T_REC**
c. **T_REC_step**
d. **T_REC_epoch**
e. **CADENCE** float, seconds, repetition interval for a sequence of repeated measurements. E.g. For AIA will be 10s for 7 channels and 30s for the other 3.
f. **DATASIGN** integer, sign of observable quantity wrt Sun center. So for observed Dopplergrams using redshift convention from observer use -1, but for heliographic remapped data use +1. Always +1 for magnetic field observations.

13. AIA lev1.5

Appendix A: Coordinate Mappings – CTYPEs

The list of CTYPE axes is at least:

- ***RAW*** Image or array with no known coordinate mapping. This will be the case for all level0 images as well as general segment arrays that have no known mappings.
- ***SOLARX*** Axis represents position East-West in arc-seconds, positive is to west, from the solar disk center. Note that this is not longitude. SOLARX is only meaningful after an image has been rotated such that the rotation axis projected into the image plane lies parallel to the Y axis. SOLARX and SOLARY will be provided for level-1.0 and above data. Normally for SOLARX and SOLARY the CRVAL will be 0.0 meaning the disk center, and CRPIX will indicate the address of the disc center. I.e. in the case that CRVAL == 0.0 then CRPIX1 == X0+1. SOLARX is deprecated in favor of HPLN-TAN.
- ***SOLARY*** Axis represent position North-South in arc-seconds from solar disk center. Note that a correction for the tilt of the rotation axis toward or away from the observer is not implied by this mapping so a positive value does not imply north of the equator. SOLARX and SOLARY usually go together and CROTA2 is usually also present. CUNIT1 and CUNIT2 default to "arcsec" if not present. SOLARY is deprecated in favor of HPLT-TAN
- ***HPLN-TAN*** Helioprojective longitude, tangent projection. Same as SOLARX but CDELT1 should be in degrees instead of arc-sec although arc-sec is allowed so long as CUNIT1 is present and contains "arcsec".
- ***HPLT-TAN*** Helioprojective latitude, tangent projection. Same as SOLARY but CDELT2 should be in degrees instead of arc-sec.
- **CRLN-CEA**, **CRLT-CEA** projections longitude vs sine-latitude
- longitude vs latitude
- Postel projections of tracked regions
- **HPLN/HPLT-TAN** cookie-cutter tracked regions based on heliographic lat/lon targets. Consider using CRPIX as cutout center instead of solar center, then CRVALs give the location of the box.

1. RAW

This would be used for level-0 data where there is no available information on the SDO roll angle at the time the images are created. Also may be used for other mappings. If CTYPE1=”RAW” there should be other keywords present to describe the data.

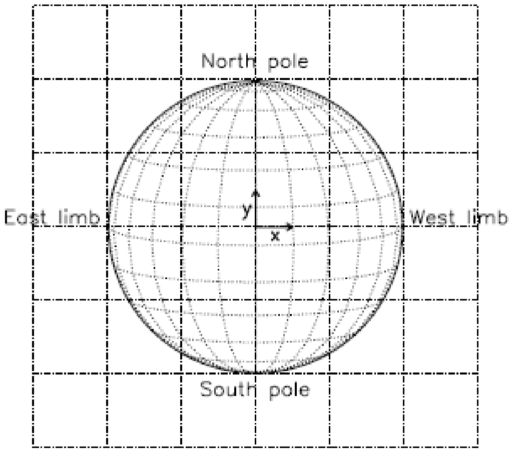


Figure A.1 Helioprojective coordinate system. (figure 2 from Thompson, 2006)

2. HelioProjective-Tangential Projections.

This is the natural system for images of the Sun made with a flat image of the sky on a flat sensor made by a perfect telescope. Distances from the image center are proportional to the tangent of the angle in the sky. An x-y coordinate system centered on the image with the x-direction being to the west on the Sun and the y-direction north on the Sun with positions measured in arc-sec is the traditional SOLARX and SOLARY system. See Figure A.1.

Here we use CTYPE1=”HPLN-TAN” and CTYPE2=”HPLT-TAN” for the longitude and latitude directions, or x, y, respectively. Assuming the image is oriented in the standard solar view with solar north at the top and solar west (west in the sky seen from northern hemisphere on the Earth) to the right. We will use units of arc-sec so CUNIT1 = CUNIT2 = “arcsec”. We will adopt the standard that the center of the solar disk is the origin so CRVAL1 and CRVAL2 both are 0.0. Then CRPIX1 is the location of disk center in the x-direction on the image and CRPIX2 is the location of disk center in the y-direction on the image.

This coordinate system presupposes that we know the telescope plate scale in arc-sec per pixel and that we know the orientation of the telescope with respect to north on the Sun. We will use the non-standard keyword IM_SCALE to indicate the telescope scale at the CCD and CDELT1 and CDELT2 for the current image scale which will differ from IM_SCALE for higher level products where the image has been rescaled.

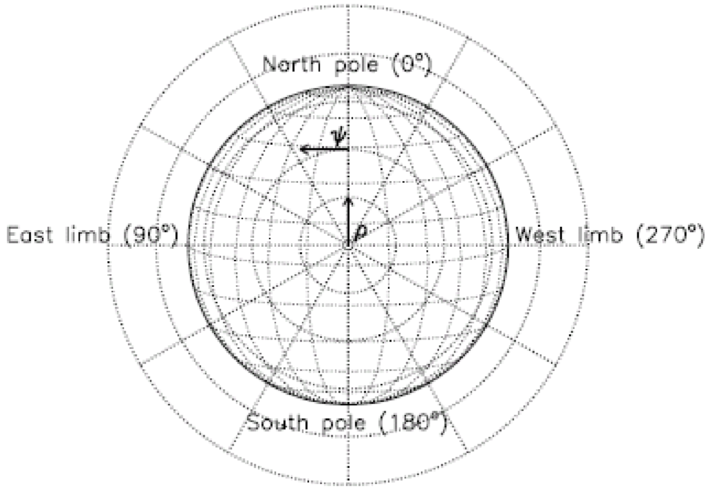


Fig. 3. A diagram of the Sun demonstrating heliocentric-radial coordinates, with lines of constant impact parameter ρ and position angle ψ overlayed. The value of ψ at each of the four compass points is also shown. The z axis points out of the page.

Figure A.2. Definition of SAT_ROT with respect to solar north. SAT_ROT is positive when SDO is rolled CCW. (Thompson, 2006)

The angular orientation of the image must be obtained from external information. For HMI and AIA this will be determined from a combination of SAT_ROT derived from the SDO “Ancillary Science Packet” and some determination of INST_ROT for each camera. SAT_ROT is the position angle of SDO Z axis measured from solar north. See figure A.2. INST_ROT is the instrument/camera offset from the SDO Z axis and is expected to change only slowly if at all. Note that this definition has the opposite sign than the normal solar position angle as used for MDI, WSO, and others based on the definition in Smart and the American Ephemeris and Nautical Almanac where “*P* *denotes the position angle of the northern extremity of the axis of rotation, measured eastward from the north point of the disk*". One system references the Sun and the other the frame of observation of the Sun. So CROTA2 = SAT_ROT+INST_ROT .But CROTA2 = -SOLARP or -P_ANGLE or other prior notations.

3. Cylindrical Equal Area CEA

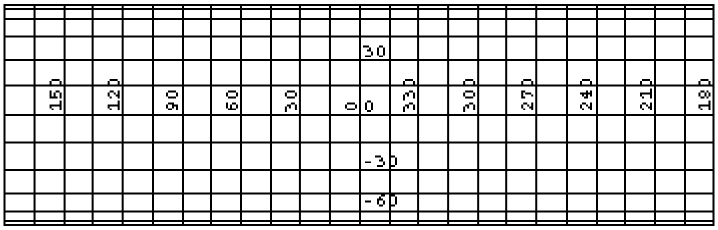


Fig. 17. Lambert's equal area projection, CEA with $\lambda = 1$; no limits.

Figure A.3, Cylindrical Equal Area – CEA – projection. From Calabretta and Greisen, 2002)

This is one of the common formats for "Carrington Synoptic Charts" where the x-direction is longitude and the y-direction is linear in sine latitude. This projection has the advantages that equal areas on the image correspond to equal areas on the Sun. For many uses this also means the per-pixel noise is nearly the same for all locations on the map. Also, any vertical strip corresponds almost to a vertical strip on some image. I.e. a crude version of such a map can be made with scissors and tape. The ratio of map width to height can be almost anything but latitude for which the mapping is conformal depends on that ratio.

Lambda	Conf latitude	ratio	examples	name
n.a.	none	3.333333	MDI Mag FITS	
1.000000	0.000	3.141593	Mt Wilson	Lambert CEA
0.954930	12.257	3.000		
0.795775	26.866	2.500	MDI mag, new Farside	
0.636620	37.071	2.000	LMSAL forecast, Yohkoh SXT	
0.572958	40.805	1.800	NSO VSM	
0.318310	55.654	1.000		

For a map of 360 degrees vs +- 1.0 a ratio of π is conformal at the equator. The relation for conformal latitude and the width/height ratio of the map ratio is $\pi * \cos^2(\theta_{conf})$ where θ_{conf} is the conformal latitude. The mapping is simple, $x=\phi$, and $y= 180*\sin(\theta)/(\pi *\lambda)$ where λ is $\cos^2(\theta_{conf})$, θ is latitude, and ϕ is longitude. The suggested keywords are:

a. CTYPE1 = "CRLN-CEA"
b. CTYPE2 = "CRLT-CEA"
c. CUNIT1="degree"
d. CUNIT2="sinlat"
e. CRVAL1 = 0 (0==360 degrees)
f. CRVAL2=0 (equator)
g. CRPIX1 is array location of 0 degrees which may be outside the array limits, remember first array location is (1,1)
h. CRPIX2 is array location of equator which may be outside the array
i. CDELT1 = degrees longitude per array element

j. CDELT2=2.0/NAXIS2 (for the NAXIS2 if the map were to include the full range +-1 of sinlat)
k. PV2_1 = λ with a default value of 1.0. A sample grid is shown in figure A.3 and a table of some example ratios in Table A.1.

CTLN-CEA defines an alternate longitude projection appropriate for both synchronic and diachronic synoptic charts and frames. CT is for Carrington Time. DATE_OBS gives the time for which a synchronic map has been constructed. PV1_n gives differential rotation coefficients used for adjustment of the longitude coordinate at a particular time. Note this is equivalent to CRLN_CEA if PV1_0 is the Carrington Rate and PV1_2,3 are 0.

The FRM_?? keywords provide information about a frame inserted into synoptic map.

a. CTYPE1 = "CTLN-CEA"
b. DATE_OBS – UT time for which synchronic mapping is done
c. CRVAL1 – Carrington Time of DATE_OBS; CT = 360*CAR_ROT – CRLN_OBS
d. CRPIX1 – array location of CRVAL1 which may be outside the array limits, remember first array location is (1,1)
e. CUNIT1 = "degree"
f. CDELT1 – degrees longitude per array element
g. CAR_ROT – Carrington Rotation number at CRPIX1 at DATE_OBS
h. CRLN_OBS – Carrington Longitude at CRPIX1 at DATE_OBS
i. PV1_0 – Solid Body Rotation Rate (nominally equatorial rate (microradians/second)
j. PV1_1 – Sine-squared coefficient of differential rotation (microradians / second)
k. PV1_2 – Sine-fourth coefficient of differential rotation (microradians / second)

l. CMIN1 – minimum Carrington Time in map
m. CMAX1 – maximum Carrington Time in map
n. CMIN2 – minimum sine-latitude of map (assuming CRLG-CEA)
o. CMAX2 – maximum sine-latitude of map (assuming CRLG-CEA)

p. FRM_MIN1 – minimum Carrington Time in frame
q. FRM_MAX1 – maximum Carrington Time in frame
r. FRM_MIN2 – minimum sine-latitude in frame (CRLG-CEA)
s. FRM_MAX2 – maximum sine-latitude in frame (CRLG-CEA)

4. Plate Carr’ee – CAR

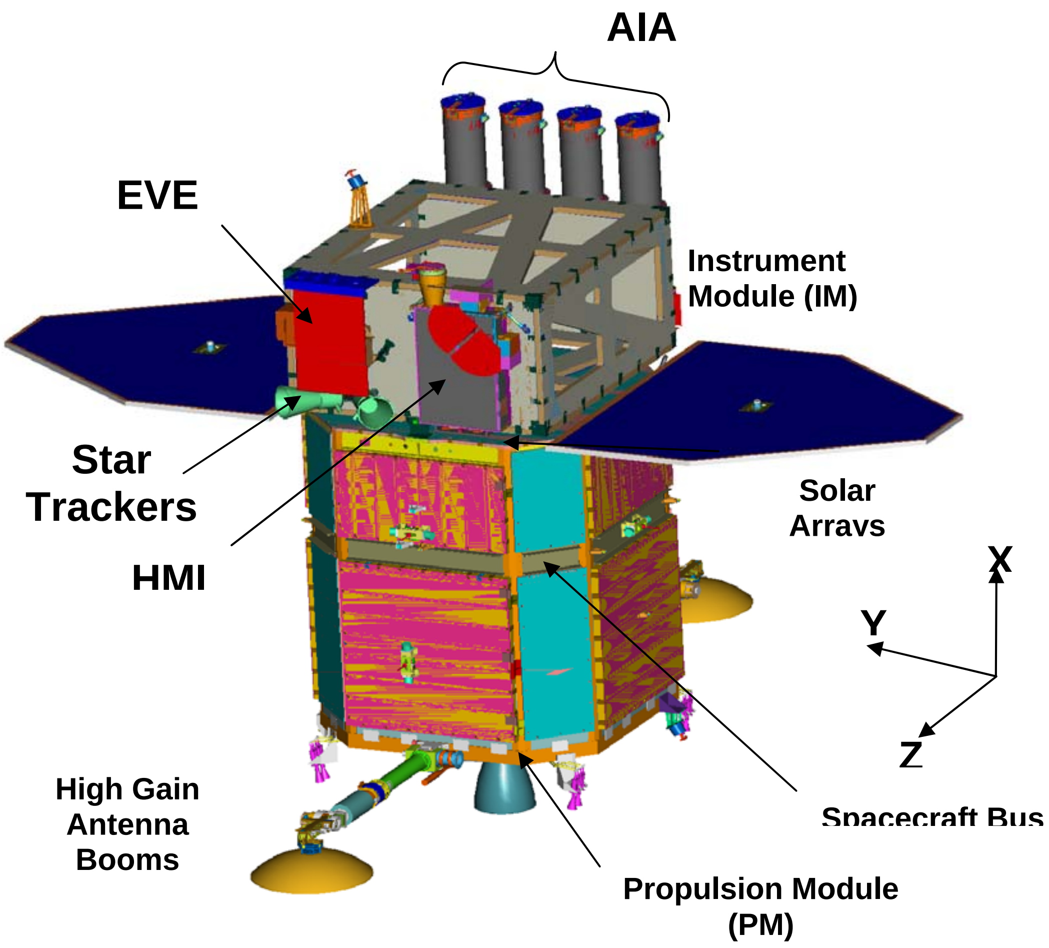
This is a simple longitude vs latitude map. The keywords are:

- **a.** **CTYPE1** = “CRLN-CAR”
- **b.** **CTYPE2** = “CRLT-CAR”
- **c.** **CRVAL1**=0.0
- **d.** **CRVAL2**=0.0
- **e.** **CRPIX1**=array location of Carrington 0 longitude
- **f.** **CRPIX2**=array location of equator
- **g.** **CUNIT1**=”degree”
- **h.** **CUNIT2**=”degree”
- i. **CDELT1**= degrees per array element
- j. **CDELT2** = degrees of latitude per array element

5. Postel

CRLT-ARC, CRLN-ARC (Azimuthal equidistant)

Appendix B. SDO Coordinate Definitions



The Sun is in the +X direction. Solar North is in the +Z direction. The annual orbit is in the –Y direction. The orbit around the Earth is counter-clockwise when looking in the –Z direction but inclined at 29 degrees from the X-Y plane.

Appendix C: Misc NOTES

SSW uses CTYPE= degrees-latitude and degrees-longitude for lat/lon heliographic cords in remapped images. The JSOC remapped image CTYPE is not yet determined but will be a WCS version.

SSW uses solar_x and solar_y and sometimes solar-x and solar-y
Some documents show solar-x and solar-y instead of SOLARX and SOLARY.
SOHO document says to use SOLARX and SOLARY. Hinode says Solar-X and Solar-Y.

The FITS Coordinate Systems for Solar Image Data WCS paper in the Helioprojective-cartesian mapping uses HPLN-TAN and HPLT-TAN with CDELTi in degrees instead of SOLARX and SOLARY with CDELT in arc-seconds.

XCEN and YCEN are a problem. They are defined in the SolarSoft documentation as:

`XCEN`	Suggested	De Facto Solar Convention	East-West FOV center of image relative to sun center in `CDELT1` units, **positive West**. `XCEN` is related to the above **FITS** keywords by: `XCEN = CRVAL1 + CDELT1 * [(NAXIS1+1)/2 - CRPIX1 ]`
`YCEN`	Suggested	De Facto Solar Convention	North-South FOV center of image relative to sun center in `CDELT2` units, **positive North**. `YCEN` is related to the above **FITS** keywords by: `YCEN = CRVAL2 + CDELT2 * [(NAXIS2+1)/2 - CRPIX2 ]`

But CRVALi only exist AFTER and CROTA have been applied so the definitions are not consistent. In particular when CROTA is not zero XCEN as built above has no meaning. A correct definition is probably:

```
a = CROTA2
XCEN = CRVAL1 + CDELT1*cos(a)*((NAXIS1+1)/2 - CRPIX1)
              - CDELT2*sin(a)*((NAXIS2+1)/2 - CRPIX2)
YCEN = CRVAL2 + CDELT1*sin(a)*((NAXIS1+1)/2 - CRPIX1)
              + CDELT2*cos(a)*((NAXIS2+1)/2 - CRPIX2)
```

STEREO also uses:

CARX_OBS	**[7]**	**Carrington X coordinate of observer**
CARY_OBS	**[7]**	**Carrington Y coordinate of observer**
CARZ_OBS	**[7]**	**Carrington Z coordinate of observer**

STEREO and AIA ask for:
OBJECT Name of object
OBJ_ID Object identifier, e.g. active region number
OBSERVER Name of observer
OBS_PROG Name of the observing program
SCI_OBJ The science objective of the observation

Appendix D: Conversion from MDI conventions for the keywords used in e.g. v2helio:

MDI	JSOC	NOTE
T_OBS	T_OBS	
OBS_B0	CRLT_OBS	STEREO uses HGLT_OBS
OBS_L0	CRLN_OBS	STEREO uses HGLN_OBS
SOLAR_P	-(SAT_ROT+INST_ROT)	Note sign difference.
IM_SCALE	IM_SCALE	Note this is CCD plate scale
S_MAJOR	n.a.	Will be fixed in lev1, use 1.0
S_MINOR	n.a.	Will be fixed in lev1, use 1.0
S_ANGLE	n.a.	Will be fixed in lev1, use 0.0
X_SCALE	CDELT1 / IM_SCALE	Rebin info, CCD pixels per bin. Implied from CDELT and IM_SCALE
Y_SCALE	CDELT2 / IM_SCALE	
DATASIGN	DATASIGN	
OBS_VR	OBS_VR	
OBS_VW	OBS_VW	
OBS_VN	OBS_VN	
ORIENT		'SESW' normal, should be set to SESW in lev0 processing.
OBS_DIST	DSUN_OBS/DSUN_REF	
R_SUN	R_SUN	
X0	X0	
Y0	Y0	
BFITZERO	n.a., use 0.0	If BFITZERO in non-zero the data should be corrected on input.

Appendix Z: FITS to DRMS to FITS keyword mapping.

We have had a number of suggestions for maintaining maps of keywords to/from various external standards. Viewed from the DRMS coder we could define a pair of routines that can do the mappings and simply use them. They can start with a simple rule and expand to do more complex mappings if and when they are implemented.

As a concrete suggestion, the following is offered as a merging of suggestions by Art and Phil. *This section will be removed or replaced in a future version of this document.*

1. Mapping functions returning pointers to static strings:
    a. char *ext_keyname(KeyTable_t *mapping, char *internal_name)
    b. char *int_keyname(KeyTable_t *mapping, char *external_name)
2. Setup function to establish mapping:
    a. KeyTable_t *get_keyname_map(char *seriesname, char *target)
    b. int free_keyname_map(KeyTable_t *mapping)
3. Mapping rules: The keyname mapping will provide a return value for both directions for any name provided.
    a. Internal to External: The function to setup a mapping will generate a mapping for all keynames in the DRMS series using default rules if specific rules are not determined by some TBD process. A name that is not an internal name will result in a null pointer returned. Default rules are:
        i. Look in the comment field of the DRMS keyname to see if the first “word” is “[*name*]” and if so, use *name*.
        ii. If the name is longer than 8 chars and contains a double, triple, or longer sequence of underscores (“__”) replace it with a single hyphen (“-“) and proceed to the next step.
        iii. Name already valid FITS name, convert to upper case and use.
        iv. Name too long, If the DRMS name is unique if truncated at 8 chars, use the truncated form. If still no solution, truncate at 7 chars and add a digit starting at 0 and increasing until a unique name is found. If that fails, truncate at 6 chars and use a 2 digit number, etc.
    b. External to internal: Mapping in this direction is easier.
        i. In the case that the series already exists then before the rule below is implemented to make the mapping table, the comment will be examined for the external name and if found, that name will be returned.
        ii. In the case that the series does not yet exist this function can be used to produce a default mapping. Since most all FITS names are OK DRMS names except for the “-“ char the default mappings need only deal with this case. The conversion will be to convert a “-“ to a sequence of “_” long enough to make the name longer than 8 chars or to 2 underscores if the name is already longer than 8 chars in case there are two or more hyphens in the name. It is expected that the new mapping generated this way will be added to the keyword comment as the initial word in square brackets when a series is made.

4. Example: with these rules the one FITS name that will be in every DRMS generated FITS file, DATE-OBS will have the DRMS name DATE__OBS.

Changes since 6 May 2009 version.

7 May 2009

1. Fix HGLN_OBS discussion to be correct.

8 May 2009

1. Fix CROTA2 to say SAT_ROT not SDO_ROLL
2. add types for Cxxxx keywords
3. remove MJD and TIME since they are deprecated and not needed by solar software.
4. add BUNIT
5. add CONTENT
6. add RECORD
7. add QUALITY
8. add DATAVALS
9. add MISSVALS
10. add TOTVALS
11. add COMMENT
12. add HISTORY

4 October 2009

1. Added QUALITY bit for place holder records in slotted series, for record slots that will never have data. Now use 0XC0000000 instead of just 0X80000000.

NEEDED
1. Postel projection keywords
2.re-arrange by topic of keywords and indicate source definition that we will try to conform to if a standard keyword.

13October 2009

Added CTYPE definition for Carrington Time longitude coordinate in Cylindrical Equal Area Projection - CTLN-CEA – Carrington Time Longitude – Cylindrical Equal Area Projection in section 3

19 Feb 2010

Rearrange most keywords into new groupings. Add table showing which level of product is expected to have which sets of keywords.

Added WCSNAME
Added BLD_VERS

Added statistics

1 April 2010

Added RSUN_REF to Observer Location set.

25 May 2010

Added SOURCE in section 3
Changed WAVELNTH to be an int.

14 Nov 2011

Fixed comment for TOTVALS to say not expected for AIA, was HMI.

16 June 2013
Modified CEA table to show lambda as well as ratio and conformal latitude.