

JSOC keywords notes – leading to a future document(s).

## **JSOC Keywords used for metadata**

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 common FITS 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 of the FITS standard. 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 both MDI and SSW 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 none 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 dataserries 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.

### **Image Coordinate Mapping Keywords**

We plan to use FITS standard keywords to describe the mapping between pixel-space and physical space, the so called “world coordinate system” or WCS. 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 CRPIX<sub>j</sub>, CRVAL<sub>i</sub>, and CTYPEx<sub>i</sub> combined with one of CROTA<sub>j</sub> and CDELT<sub>i</sub>; PC<sub>i</sub>\_j<sub>s</sub> and CDELT<sub>i</sub>; or just CD<sub>i</sub>\_j<sub>s</sub> where in the common usage of e.g. CTYPEx<sub>1</sub>=SOLARX 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 a crude estimate of this relationship is sufficient. For helioseismology it is 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.

It is however likely that for many users who are interested primarily in morphology that a rough approximation of the telescope image scale and distortions is sufficient for quick-look use. We propose to allow increasing accuracy of the mapping keywords as the data progresses from as-

observed telemetry data to higher levels of analysis. In this process we will explicitly specify what the approximations are and which keyword contains measured information and which contain derived quantities. This will involve maintaining some “legacy” keyword names that have been developed to accurately describe the data at lower levels of processing and transitioning into the WCS keywords as the preferred usage at higher levels of calibration.

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-0.3) CTYPE will be set to SOLARX and SOLARY, with CROTA1 == CROTA2 to be logically applied *before* assigning the implied labels of X and Y used to map array index 1 and 2. The “SOLARY” direction will be the projection of the Carrington solar rotation axis onto the plane of the sky (+ is north) and “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”, “X0”, and “Y0” will contain the key information from which the other values are computed. R\_SUN, X0, Y0 are all in pixels with center of the lower left pixel of the array set to 0.0, 0.0. X0 and Y0 are the location of the solar disk center in the image as is. Then we probably set CRPIX1 == X0+1, CRPIX2 == Y0+1.

Then in order:

From SDO attitude and orbit information:

**SAT\_ROT** = for lev<0.5, intended/commanded 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 convention must be verified with SDO FDS conventions.). For lev>=0.5 this will be corrected with data 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.

**WAVELNTH** wavelength (nm) of observation. For HMI use 617.3.

For each CCD camera:

**INST\_ROT** = Telescope roll angle. The angle between the instrument CCD Y axis and the SDO Z axis. This is a calibration (nearly constant) determined for each AIA ATA or HMI camera. 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, starting at level-0 we can provide the following:

For AIA:

**X0, Y0** = computed from commanded pointing and known offsets, pixel address of the science reference boresite with telescope specific corrections.

**IM\_SCALE** Predefined AIA plate scale in arc-seconds per pixel.

**R\_SUN** computed from DSUN\_OBS, RSUN\_REF, and IM\_SCALE.

For HMI:

**X0, Y0** for lev<0.8 computed from commanded pointing and known offsets, pixel address of the science reference boresite with telescope specific corrections; for lev >= 0.8 computed from fit to image

**R\_SUN** for lev<0.8 computed from IM\_SCALE, DSUN\_OBS and RSUN\_REF; for lev>=0.8 fit to image.

**IM\_SCALE** for lev<0.8 set to ~ 0.5; for lev >=0.8 computed from R\_SUN and DSUN\_OBS and RSUN\_REF.

For both AIA and HMI for lev0.3 and above:

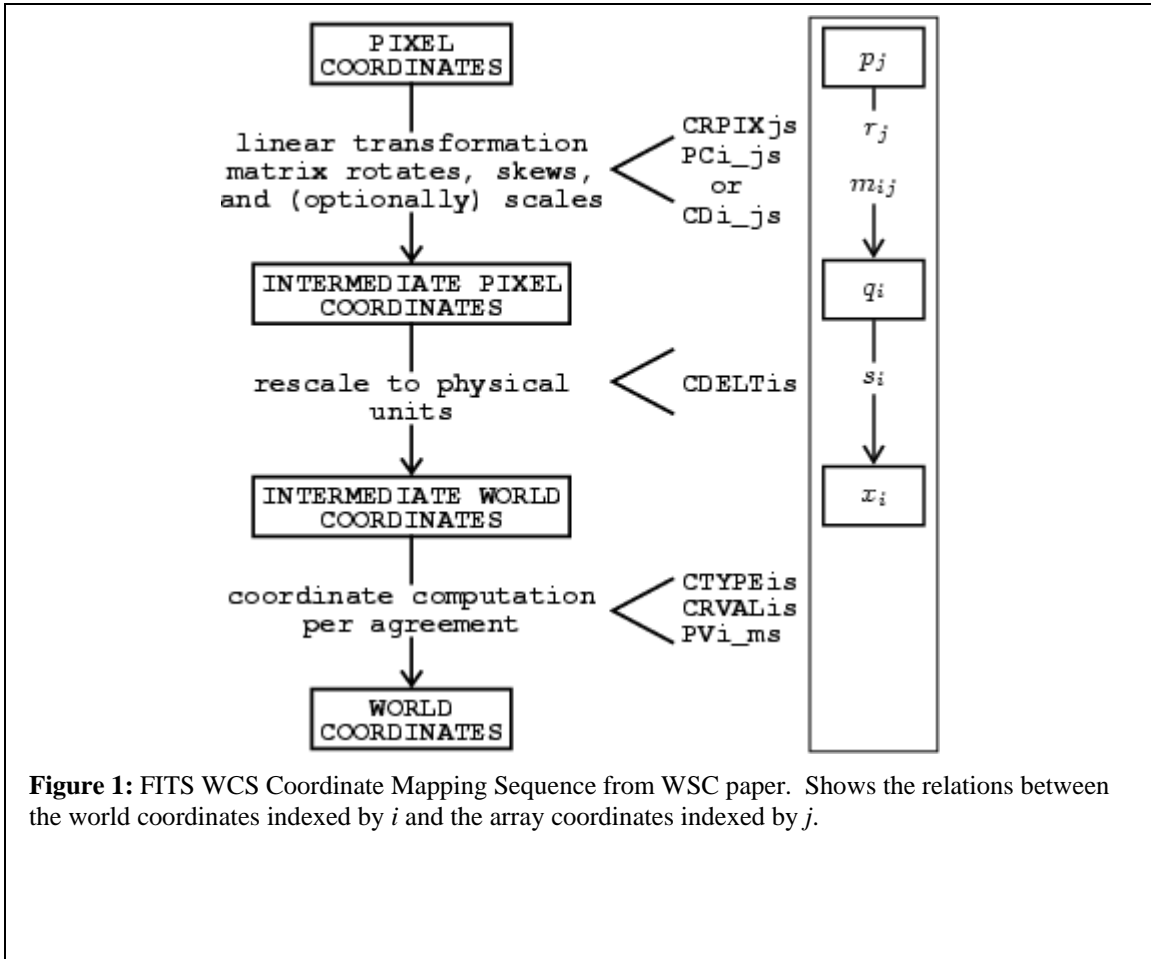
**CDELTA1 == CDELTA2** = IM\_SCALE if full resolution, else scaled from IM\_SCALE.

**CROTA1 == CROTA2** = SDO\_ROLL with corrections level 0.5 and above.

**CRPIX1, CRPIX2** = X0, Y0

**CRVAL1 = CRVAL2** = 0.0,0.0

**XCEN, YCEN** computed from above.



## General Descriptive Keywords in DRMS and for FITS Exports

There are a number of general purpose keywords that are FITS standard reserved keywords that can be adopted as DRMS keywords. These are NOT all present at all levels of processing but will only be added when they can be properly determined. They will not be carried to higher levels when they are no longer applicable. In the list below when the DRMS name is not a valid FITS name, the FITS equivalent name follows a “/”. These keywords fall into several sets:

1. FITS required keywords: These do not appear as DRMS keywords since they are used for the basic structure of the stored data which 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
  
2. FITS Reserved keywords for source of data. These will be generated for HMI and AIA data for all records.
  - 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 UTC.
  - b. **DATE-OBS** text, Date and time when observation of this image was started. Uses DATE format.
  - c. **DATE\_OBS** old SOHO keyword is *not present, needed or used for SDO*. Differs from DATE-OBS by less than 0.1 second for the SDO orbit. Was c. 5 seconds for SOHO.
  - d. **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. Suggestion is to use either: “SDO/HMI” or “SDO/AIA” as possible values.
  - e. **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 might be: ATA\_1, ATA\_2, ATA\_3, ATA\_4, FRONT, SIDE or similar choices. We should adopt and stick with a set of names. INSTRUME contains the same information as the integer value in the CAMERA keyword.
  - f. **WAVELNTH** float, Wavelength of observation in nm. This keyword will be a constant, 617.3 for HMI and one of **XXXX** for AIA. It will NOT be set to a higher accuracy for each HMI filtergram.
  - g. **ORIGIN** text, Location where file was made. e.g. “SDO/JSOC-SDP”.
  
3. Image coordinate mapping keywords, FITS standard. Specifies mapping from array axes (*j*) to image axes (*i*).
  - a. **CTYPEi** text, specifies type of image axis *i* for other Cxxxx keywords.
  - b. **CRPIXj** Reference pixel for array axis *j*. First pixel is number 1 (not 0).
  - c. **CRVALi** Physical value for image axis *i* at the center of the pixel.
  - d. **CDELTi** Physical increment per index value for image axis *i*

- e. **CUNIT*i*** Physical units for positions on image axis *i*.
- f. **CROTA*j*** Rotation needed for array axes to get to image axes. Unit is degrees and since all **CROTA*j***s are the same for cases where we will use this notation only the first axis need be specified. CROTA 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).
- g. **CRDER*i*** Estimate of random error in coordinate *i* expressed in CUNIT*i*.
- h. **CSYSER*i*** Estimate of systematic error in coordinate *i* expressed in CUNIT*i*.

These Cxxxxx keywords may have multiple sets present. If so the sets beyond the first have a single letter suffix 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.

4. SSW keywords. These keywords are expected by SolarSoft IDL code and may as well be provided on export. They are redundant with other keywords so probably need not be in the DRMS list. The "probably" here questions if we want e.g. XCEN to be there for DRMS via SSW use.

- a. **MJD** float, Date of observation as modified julian day.
- b. **TIME** float, Time of observation in seconds within a day. MJD and TIME describe the same instant as DATE-OBS.
- c. **XCEN** Location on the Sun of the center of the field of view in the array in CDELTA1 units (usually arc-sec). Only makes sense for reduction levels where CTYPE1 is "SOLARX". Is computed from  $a = \text{CROTA2}$ ;  $\text{XCEN} = \text{CRVAL1} + \text{CDELTA1} * \cos(a) * ((\text{NAXIS1} + 1) / 2 - \text{CRPIX1}) - \text{CDELTA2} * \sin(a) * ((\text{NAXIS2} + 1) / 2 - \text{CRPIX2})$
- d. **YCEN** Is computed from  $a = \text{CROTA2}$ ;  $\text{YCEN} = \text{CRVAL2} + \text{CDELTA1} * \sin(a) * ((\text{NAXIS1} + 1) / 2 - \text{CRPIX1}) + \text{CDELTA2} * \cos(a) * ((\text{NAXIS2} + 1) / 2 - \text{CRPIX2})$

5. DRMS keywords. These keywords have various heritages.

- a. **CAMERA** integer describing the camera used, AIA: 0-3, HMI: 0-1.
- b. **FSN** integer containing "Frame Sequence Number" or "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.
- c. **EXPTIME** float, seconds, Exposure duration. Often  $\text{T\_OBS} = \text{DATE-OBS} + \text{EXPTIME} / 2.0$ .
- d. **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 sprint\_ut 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.

- e. **T\_PACKET** time (double) version of telemetry packet time in TAI. Converted to text and shown as UT on export or when printed.
- f. **T\_REC** time (double) when internal and text in UT when exported or printed. The “nominal” slot time for a record containing an “observable” observed at a regular cadence using the DRMS prime key time\_slot tools. Will usually differ from T\_OBS by less than half the cadence. Will be computed from the cadence (T\_REC\_STEP) and “slot” number from the reference epoch T\_REC\_EPOCH. The keyword T\_REC\_TYPE must contain TS\_EQ. Makes sense to provide for level1 and above when equally spaced time series. T\_REC will be the primary index for most HMI level-1 data products.
- g. **T\_REC\_STEP/T\_STEP** time (double) time in seconds between adjacent measurements. Makes sense to provide for level1 and above when equally spaced time series.
- h. **T\_REC\_EPOCH/T\_EPOCH** time (double) reference epoch in TAI internally or UT externally or exported. The reference epoch used for T\_REC times. Defaults to 1976.12.31\_23:59:45.000\_UTC which is 0.0 for the DRMS internal time.
- i. **T\_REC\_TYPE/T\_TYPE** (string) contains TS\_EQ for equal spaced time slots.
- j. **DSUN\_OBS** float, Distance from Sun’s center to SDO in m.
- k. **R\_SUN** float, Radius of the Sun’s image in pixels.
- l. **RSUN\_REF** float, Radius of Sun in m, depends on wavelength.
- m. **CRLN\_OBS** float, Carrington longitude of observer, same as SOI **OBS\_L0**.
- n. **CRLT\_OBS** float, Carrington latitude of observer, same as MDI **OBS\_B0**.  
Note that STEREO uses HGLN\_OBS which is the longitude from disk center, i.e. the Stonyhurst longitude. This seems too intertwined with the T\_OBS with possible errors keeping the sign and rotation correct when  $|CRLT\_OBS| > 90$ . So we recommend CRLT\_OBS which does not need a simultaneous value of T\_OBS to interpret.
- o. **CAR\_ROT** integer, Carrington rotation number of CRLN\_OBS. This is the same as MDI **OBS\_CR**.
- p. **SAT\_ROT** float, degrees, position angle of solar pole wrt the SDO Z axis. This value is required as input to the CROTA calculation. CROTA will also need the particular instrument offset (hopefully fixed) from the SDO Z axis.
- q. **INST\_ROT** float, degrees, rotation of the camera from the SDO Z axis.  
**CROTA<sub>j</sub>** will be the sum **SAT\_ROT + INST\_ROT**. With this definition **CROTA<sub>j</sub>** is the negative of MDI **SOLAR\_P**.
- r. **X0** float, x-axis location of solar disk center in pixels, start 0.0.
- s. **Y0** float, x-axis location of solar disk center in pixels, start 0.0. For HMI X0 and Y0 are added at level-0.8 and 0.9 and combined with R\_SUN are the source information to compute the values for the coordinate information from this level and higher. For AIA these quantities will be computed *from* the WCS keywords. X0 and Y0 wrt the center of the lower left pixel.
- t. **IM\_SCALE** float, arc-sec per CCD pixel default value for the particular instrument. This value will be used for the estimate of CDELTA for AIA and will



be updated from R\_SUN and D\_SUN for HMI at and above level-0.8. Note that IM\_SCALE does not change when the image is rebinned on the ground. It is a characteristic of the data as observed.

- u. **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.
- v. **OBS\_VR** float, speed of observer in radial direction, + is away from Sun.
- w. **OBS\_VW** float, speed of observer westward, + in direction of Earth orbit.
- x. **OBS\_VN** float, speed of observer northward (heliographic), + to north.

## Coordinate Mappings – CTYPEs

The list of CTYPE axes is at least:

1. **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.
2. **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 used for level-0.3 and above data.
3. **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 CROTA1 is usually also present. CUNIT1 and CUNIT2 default to “arcsec” if not present.
4. **HPLN-TAN** Helioprojective longitude, tangent projection. Same as SOLARX but CDELTA1 must be in degrees.
5. **HPLT-TAN** Helioprojective latitude, tangent projection. Same as SOLARY but CDELTA2 must be in degrees.
6. projections longitude vs sine-latitude or longitude vs latitude
7. Postel projections of tracked regions
8. **TBD** cookie-cutter tracked regions based on heliographic lat/lon targets. In these cases the mapping is same as SOLARX/Y but use of lat/lon is more natural as the reference values.

## Coordinate Mappings for level 0.3 and above for HMI and AIA

The FITS WCS standard allows multiple coordinate specifications to be present. This would allow e.g. both a SOLARX/Y and HPLN-TAN/HPLT-TAN specification to be given. In this case CTYPE1A would be HPLN-TAN, etc. Therefore we recommend that two sets be used for level 0.3 and above. The unadorned first set (i.e. no suffix letter) will be **SOLARX** and **SOLARY** and the second set, e.g. **CTYPE1A** and **CTYPE2A** will be **HPLN-TAN** and **HPLT-TAN** respectively. Also then we should set WCSNAME = ‘solarxy’ and WCSNAMEA = ‘Helioprojective-cartesian’

**Conversion from MDI conventions for the keywords used by e.g. v2helio:**

<b>MDI</b>	<b>JSOC</b>	<b>NOTE</b>
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	CDELTA1 / IM_SCALE	Rebin info, CCD pixels per bin. Implied from CDELTA and IM_SCALE
Y_SCALE	CDELTA2 / 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	
XSCALE		
X0	X0	
Y0	Y0	
BFITZERO	n.a., use 0.0	If BFITZERO is non-zero the data should be corrected on input.

**Misc NOTES**

SSW uses CTYPE= degrees-latitude and degrees-longitude for lat/lon heliographic cords in remapped images.

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.

And add a second set, e.g. CTYPE1A = HPLN-TAN with CDELTA1A in degrees, etc. for the export fixup.

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

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

<b>XCEN</b>	Suggested	De Facto Solar Convention	East-West FOV center of image relative to sun center in CDELTA1 units, <b>positive West</b> . XCEN is related to the above <b>FITS</b> keywords by: $XCEN = CRVAL1 + CDELTA1 * [(NAXIS1+1)/2 - CRPIX1]$
<b>YCEN</b>	Suggested	De Facto Solar Convention	North-South FOV center of image relative to sun center in CDELTA2 units, <b>positive North</b> . YCEN is related to the above <b>FITS</b> keywords by: $YCEN = CRVAL2 + CDELTA2 * [(NAXIS2+1)/2 - CRPIX2]$

**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:**

$a = CROTA2$

$XCEN = CRVAL1 + CDELTA1 * \cos(a) * ((NAXIS1+1)/2 - CRPIX1)$   
 $\quad - CDELTA2 * \sin(a) * ((NAXIS2+1)/2 - CRPIX2)$

$YCEN = CRVAL2 + CDELTA1 * \sin(a) * ((NAXIS1+1)/2 - CRPIX1)$   
 $\quad + CDELTA2 * \cos(a) * ((NAXIS2+1)/2 - CRPIX2)$

**LONPOLE, LATPOLE, WCSNAMEa**

**DSUN\_OBS instead of D\_SUN**

**HGLN\_OBS =0 and HGLT\_OBS=B0**

**Or CRLN\_OBS is Carrington longitude of disk center.**

**STEREO 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**

**CAR\_ROT [5] Carrington rotation number**

**OR CRLN\_OBS and CRLT\_OBS**

**RSUN\_REF** [7] **Value of R<sub>sun</sub> (meters) used in determining coordinates**

**ORIGIN** [1] **Responsible organization or institution**

STEREO and AIA ask for:

*OBJECT* Name of object

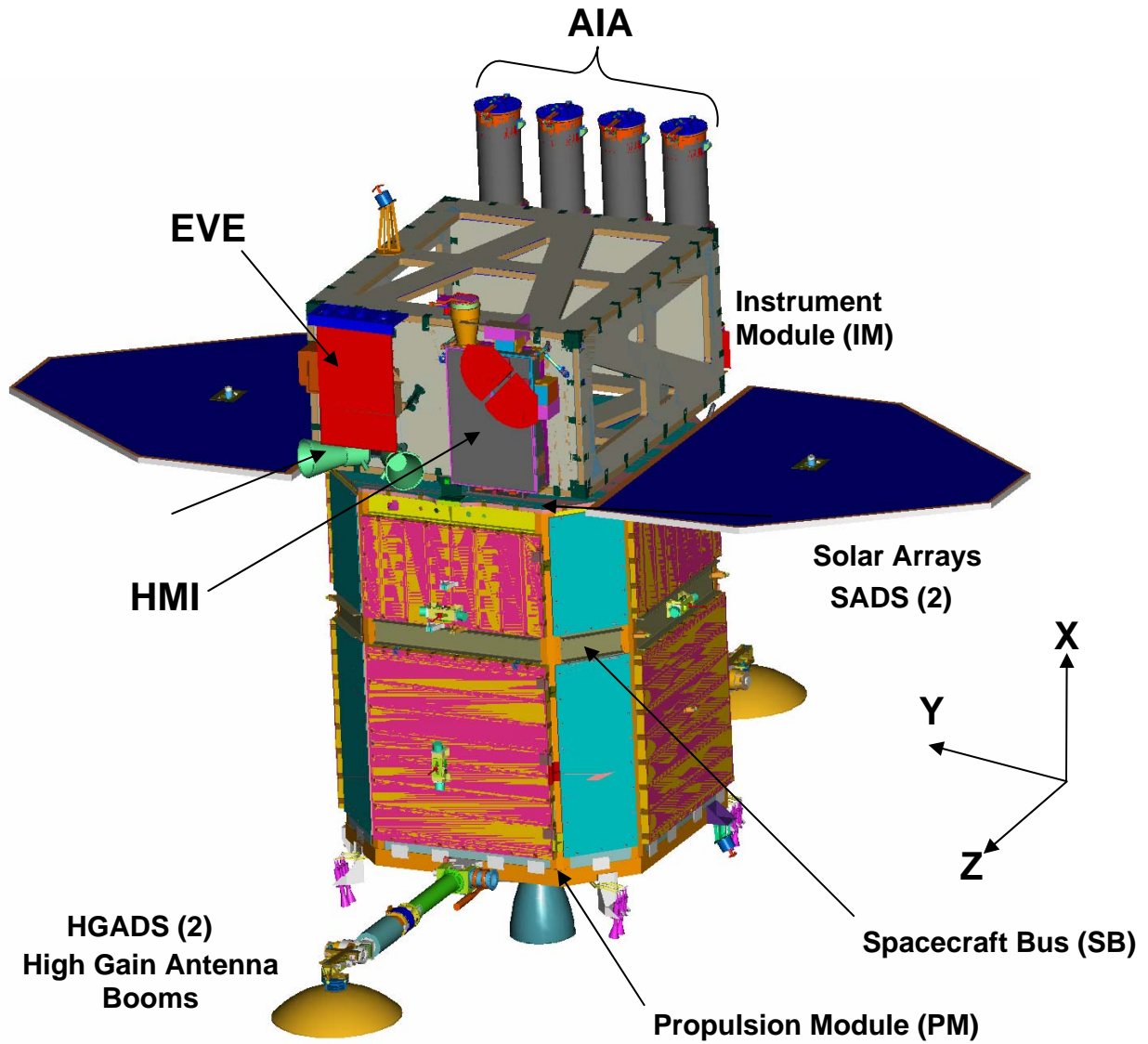
*OBJ\_ID* Object identifier, e.g. active region number

*OBSERVER* Name of observer

*OBS\_PROG* Name of the observing program

*SCI\_OBJ* The science objective of the observation

Appendix A. SDO Coordinate Definitions



## Appendix B: 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.