CORONAL DIAGNOSTIC SPECTROMETER

SoHO

CDS SOFTWARE NOTE No. 11

Version 1.0 28 March 1994

CONVERTING CDS CALIBRATION TELEMETRY TO FITS FILES

W. Thompson
Applied Research Corporation
NASA Goddard Space Flight Center
Laboratory for Astronomy and Solar Physics
Code 682.1
Greenbelt, MD 20771, USA

William.T.Thompson.1@gsfc.nasa.gov
pal::thompson

1 Overview

The software for converting telemetry taken during the calibration of CDS is modelled after that planned for operations. However, there are a number of significant differences that are a result of the different environments during test and during operations. These differences are outlined in this document.

One of the major differences between calibration and operation is the lack of the science and technical planning databases. During operations, the FITS formatting software will use these databases to annotate the data. These databases will not be available during the CDS calibration effort.

Another major difference concerns the order in which the data are written out. During calibration it is planned to write out the data every exposure, to minimize the amount of data that needs to be stored in memory. It was planned that during operations the data would be written out only at the completion of a raster. (However, see Section 7 below.)

The keywords used in this software reflect a change from those used previously. Before the keywords used for pointing were based on an older draft of the SoHO Science Operations Plan (SOP) which referred to a YZ coordinate system on the apparent solar disk. The latest version of the SOP uses an XY coordinate system, which is more usual and matches the CDS internal documentation. Thus, for instance, instead of having TDETYn and TDETZn, we have TDETXn and TDETYn. All FITS keywords should be considered to be provision at this stage, however.

2 Routines

The sequence of events needed to write out the calibration data into FITS files is as follows:

1. First, the routine OPEN_CALFITS is used to initialize the FITS headers, and set up the structure of the binary table. The calling sequence for this routine is

   ```
   OPEN_CALFITS, FILENAME, DETECTOR, RASTERHDR, CALHDR  [, PATH=PATH ]
   ```

   DETECTOR is either the character string "NIS" or "GIS" depending on which spectrometer took the data. RASTERHDR will be either VDSRASTERHDR or GISRASTERHDR.

2. Next, the routine WRITE_CALFITS is called for each exposure. The FITS file is not actually created until the first time WRITE_CALFITS is called. The calling sequence is

   ```
   WRITE_CALFITS, DETECTOR, EXPHDR, EXP_ARRAY
   ```

3. Finally, the routine CLOSE_CALFITS is called to close the file. It has no parameters.

The routine READ_CALFITS is used to read in the data file. Its use is analogous to READ_CDS. For example, to read in the data from the first data extraction window, one would use the following commands:

```
FXBOPEN, UNIT, 'testcaldata.fits', 1     ;Opens the table
READ_CALFITS, UNIT, DATA, 'WINDOW1'      ;Reads WINDOW1 data
FXBCLOSE, UNIT                           ;Closes the table
```

3 Primary and binary table headers

The keywords that will be used in the main and binary table headers are listed in Table 1. A number of keywords planned for operations have been left out. Most were left out because they would depend on information taken from the science planning database, which will not be available during calibration. Others are simply not applicable outside of actual operations.

One keyword that was left out was the end date/time of observation, "DATE_END". This was left out because it could only be determined once all the data had been received, at which point the FITS headers had already been written to the file. However, there is a way to include this keyword, if desired during calibration. One does this by leaving space for it in the headers, and then when closing the file, rewriting the updated headers directly over the original headers.

The keywords "WAVEMIN" and "WAVEMAX" were also left out, under the philosophy that one won't really know what the proper values for these will be until after calibration.

4 Data stored in binary table columns

The data windows will be stored in binary table columns labeled "WINDOW1", "WINDOW2", etc. for lack of any better naming convention. During operations, the names of these columns will presumably be based on information from the science and technical planning databases, and should be more informative. Other table columns are listed in Table 2. Omitted are the columns that would give the absolute pointing of the instrument based on the other information. Again, this is because calibration measurements need to be made before this can be done properly.

The proposed format for the FITS files during operations included a number of descriptive parameters for the data window and auxilliary data columns. Most of these are omitted for the calibration data, except for TDETXn and TDETYn (the latter is only used for NIS data) which give the starting position of the data windows, and TUNITn and TDESCn which describe the units and dimensions of the data. If desired, the other descriptive keywords which have been left out could be reinstated.

The omitted keywords TRPIXn, TRVALn, and TDELTn are used to describe how the data relate to physical quantities—e.g. arcseconds on the Sun—and require calibration before they can be properly calculated. The same comment applies to the keywords TWAVEn, TWMINn, and TWMAXn. TNULLn was omitted as being inappropriate—the SCI_ANA software doesn't handle missing data in the same way as the FITS formatting software used during operations will. The keywords TDMINn and TDMAXn cannot be calculated until after all the data have been received. If necessary, they can be added using the technique described above for DATE_END (Section 3).

5 Partially resolved issues

Some issues which have been at least partially resolved. Those that merit mention are as follows:

- I have had input from Martin Carter as to what the data types are of the various 16 bit parameters in the telemetry. Thus, the following assumptions are made:

Table 1: Keywords to be used in the FITS primary and binary table header for CDS data taken during calibration.

FILENAME	Name of data file (no path or extension)
DATE	Date file was written
ORIGIN	The character string "RAL"
TELESCOP	"CALIBRATION"
INSTRUME	"CDS"
DETECTOR	Either "NIS" or "GIS"
PROG_ID	From SCIHEAD.STUDYID
PROG_NUM	SCIHEAD.STUDYCOUNTER
SEQ_IND	SCIHEAD.RASTERCOUNTER
DATE_OBS	From first exposure header
DATACOMP	SCIHEAD.DATACOMPSCHEME
COMP_OPT	SCIHEAD.DATACOMPOPTION
WNDW_ID	SCIHEAD.DATAEXTWINDOWID
SER_ID	SCIHEAD.SEQUENCEID
EXPCOUNT	SCIHEAD.RASTERLENGTH
EXPTIME	From SCIHEAD.EXPOSURETIME, converted to fractional seconds
OPS_L	SCIHEAD.OPSLCOORDINATE
OPS_R	SCIHEAD.OPSRCOORDINATE
SLIT_POS	SCIHEAD.SLITPOSITION
SLIT_NUM	SCIHEAD.SLITNUMBER
MIR_POS	SCIHEAD.MIRRORPOSITION
EV_ENAB	SCIHEAD.SCIPROCSTATUS: Event enabled
COMP_ERR	SCIHEAD.SCIPROCSTATUS: Data compression scheme error
VDS_HTR	SCIHEAD.SCIPROCSTATUS: VDS high time resolution mode
VDS_ORI	SCIHEAD.SCIPROCSTATUS: VDS readout orientation
VDS_ACC	SCIHEAD.SCIPROCSTATUS: VDS accumulate mode
VDS_MODE	SCIHEAD.SCIPROCSTATUS: VDS camera readout mode
VDSBINX	Binning factor in X
VDSBINY	Binning factor in Y
GAS_ID	CALHDR.GAS_ID
GAS_VOLT	CALHDR.GAS_VOLTAGE
GAS_CURR	CALHDR.GAS_CURRENT
GAS_PRES	CALHDR.GAS_PRESSURE

Table 2: Auxilliary information to be stored in columns within the binary table.

DEL_TIME	Time offsets from start of observation
OPS_L	EXPHDR.OPSLCOORDINATE
OPS_R	EXPHDR.OPSRCOORDINATE
SLIT_POS	EXPHDR.SLITPOSITION
MIR_POS	EXPHDR.MIRRORPOSITION
PITCH	EXPHDR.SUNSENSORPITCH
YAW	EXPHDR.SUNSENSORYAW
EV_LEVEL	EXPHDR.EVENTLEVEL
SOLAR_X	EXPHDR.SOLARX
SOLAR_Y	EXPHDR.SOLARY
EV_RECOG	EXPHDR.EVENTSTATUS: Event recognition enabled
EV_DETEC	EXPHDR.EVENTSTATUS: Event detected
EV_VALID	EXPHDR.EVENTSTATUS: Solar coordinates valid

Raster header packets		
Signed	Unsigned	"MCU telemetry"
StudyCounter	DataCompOption	OPSLCoordinate
RasterCounter		OPSRCoordinate
DataExtWindowID		SlitPosition
SequenceID		SlitNumber
RasterLength		MirrorPosition
ExposureTime		

Exposure header packets		
Signed	Unsigned	"MCU telemetry"
StudyCounter	EventLevel	OPSLCoordinate
RasterCounter		OPSRCoordinate
ExposureNumber		SlitPosition
SolarX		MirrorPosition
SolarY		
SunSensorPitch		
SunSensorYaw		

The reasons why one parameter is signed and another unsigned is not clear. At the moment, I do not have a definition of the MCU telemetry format.

- The "Sequence ID" in the telemetry is now recognized to be a CDHS-specific concept distinct from (but related to) the raster definition. To make this distinction clear, it is now referred to as the "on-board sequence load ID" in the FITS header.
- Since the definition of the science process status word is in a state of flux, it was decided to simply write this parameter out as a character string of ones and zeros.

6 Unresolved issues

Some issues need to be resolved before the software can be considered to be completed:

- Currently the SCI_ANA software deselects the data extraction windows into full detector arrays. The CALFITS procedures are written assuming that the data is instead passed as an array with the dimensions

```
(LENGTHX, LENGTHY, N_WINDOWS)   ;NIS
(LENGTH, N_WINDOWS)             ;GIS
```

  This is so that SCI_ANA can be upgraded to allow for more than 10 exposures in a raster. However, this part of the routine is commented out, and additional code is inserted to make the software work with the current version of SCI_ANA.

- The software is written to expect that the exposure start time in the exposure header is passed as a single six byte array variable called "ExposureStart" instead of the three variables "ExposureStartMS", "...NMS", and "...LS" which the routine SCI_ANA uses now. This simplification should be a simple change to make, since this variable currently only occurs in two places, namely SCI_ANA and FMT_CDHS_TIME.

- There is no parameter in the telemetry to describe the overall format of a raster. For example, if a GIS raster scan involves 10 X positions by 10 Y positions, then the parameter SCIHEAD.RASTERLENGTH will be 100, but nowhere in the telemetry does it explicitly state that those 100 exposures are subdivided into a 10×10 pattern. It could, perhaps, be deduced from the other parameters, i.e. the slit and mirror positions, etc., but this could possibly be muddied by other effects such as feature tracking. During operations, it was planned to derive this information from the science plan. How should this be handled during calibration—should we just assume it will be evident from the slit and mirror positions?

- A number of items are listed as being in "MCU telemetry" format. At present I do not know exactly what this means. I will need to know more about this before I can assess whether or not the data should be left in this format, or converted to something else before putting in the FITS files.

  Values such as the mechanism positions or sun sensor values should be retained as raw values in the FITS files. Any calibrations that need to be applied should be done when the file is read. However, the file should contain valid numbers, not bit codes. Any conversions that don't rely on calibrations should be applied before the data is written to the FITS file.

  In the meantime, the relevant data is written as is, and annotated as being in "MCU telemetry" units. If the software is later modified to convert these numbers into some other format, then the value of the TUNITn keywords can be changed to reflect this.

7 Implications for operations

Writing this software for the CDS calibration effort is a good testing ground for writing the software for CDS operations. However, one significant difference between operations and calibration is the sequence in which the data is written out. During operations it was planned to write the data out in one chunk at the end of each raster. The advantages perceived for this approach are:

- Having all the data in a single row in the FITS binary table seemed a simple approach that matched the requirements of both the NIS and GIS data. Writing each exposure out to a separate row matched the needs of NIS data well, but not GIS data.
- The order of the data dimensions can be modified to better fit the science requirements, before writing out to the file.
- Once all the data is read in, it can be easily decided if the data can be written out as short integer, or needs to be treated as long integer, before writing any data out to the file.

The shortcomings of this approach are that:

- All the data needs to be kept in memory simultaneously. A typical raster should only come to a few megabytes total, so this shouldn't be a problem. However, there may be extreme cases, particularly in a "sit-and-stare" mode, that might strain the limits of the system.
- A large part of the processing is done at the end of a raster. This can lead to a certain amount of delay, during which time telemetry packets will need to buffered by the CDS telemetry distribution software before IDL is ready to start accepting them again. It's unclear to me what the limits are on the amount of buffering that can be handled.

One possible option that I can see is to possibly combine these two approaches. A raster definition could contain a parameter which corresponds to "write the data out to disk every N exposures". For normal NIS and GIS rasters, N would be equal to the size of the raster. For "sit-and-stare" operations, N could be 1. Other possibilities could also be defined on a case-by-case basis.