CORONAL DIAGNOSTIC SPECTROMETER
SoHO

CDS SOFTWARE NOTE No. 9

Version 2.0 21 November 1996

THE CDS QUICKLOOK DATA STRUCTURE (QLDS)

Stein Vidar Hagfors Haugan
Institute of Theoretical Astrophysics
University of Oslo
PO Box 1029
Blindern, 0315 Oslo
Norway

s.v.h.haugan@astro.uio.no

A brief description of the QuickLook Data Structure (QLDS) used within the CDS Quicklook software project.

The function READCDSFITS returns the information of a CDS fits file as a single anonymous structure with the following tags:

```
{
; Tags for internal use by the QuickLook software.

 QL_ID,         ; The ID of the current QLMGR
 QL_NO,         ; The registration number within the QLMGR

; Header information

 HDRTEXT        ; A string array with the original FITS header
 HEADER         ; Structure w/values of FITS header parameters (see below)

; Detector data descriptors

 DETDESC        ; Array of structures, descriptors for data in each of the
                ; detector data windows (line windows)

; Background counts/wavelength calibration

 BACKGROUND     ; A structure w/information from the background (bias)
                ; readout windows.
 WAVECAL        ; The wavelength calibration structure to be used with
                ; this data. You may modify this or assign to it another
                ; wavelength calibration (see the routine GET_WAVECAL),
                ; if you like

; Auxiliary data:

 DEL_TIMEDATA   ; DEL_TIME data for each exposure (time in seconds since
                ; first exposure).
 DEL_TIMEDESC   ; A descriptor for the data in DEL_TIMEDATA

 INS_XDATA      ; INS_X (pointing) data
 INS_XDESC      ; Descriptor for INS_X data array.

 INS_YDATA      ; INS_Y (pointing) data
 INS_YDESC      ; Descriptor for INS_Y data.

 OPSLBITSDATA   ; OPS (Offset Pointing System) Bits Left data.
 OPSLBITSDESC   ; Descriptor

 OPS_LDATA      ; OPS Left Data
```

```
OPS_LDESC     ; Descriptor

OPSRBITSDATA  ; OPS Bits Right data
OPSRBITSDESC  ; Descriptor

OPS_RDATA     ; OPS Right data
OPS_RDESC     ; Descriptor

SLITBITSDATA  ; Slit data
SLITBITSDESC  ; Descriptor

SLIT_POSDATA  ; Slit position data
SLIT_POSDESC

MIR_POSDATA   ; Mirror position data
MIR_POSDESC

PTCHBITSDATA  ; PITCH Bits data
PTCHBITSDESC

PITCHDATA     ; PITCH data
PITCHDESC

YAW_BITSDATA  ; YAW Bits data
YAW_BITSDESC

YAWDATA       ; YAW data
YAWDESC

EV_LEVELDATA  ; Event level data
EV_LEVELDESC

ONBOARDXDATA  ; Onboard X data
ONBOARDXDESC

ONBOARDYDATA  ; Onboard Y data
ONBOARDYDESC

EV_RECOGDATA  ; Event recognition data
EV_RECOGDESC

EV_DETECDATA  ; Event detection data
```

```
 EV_DETECDESC

 EV_VALIDDATA  ; Event valid data
 EV_VALIDDESC

 BKGRCDDATA    ; Background data (GIS only)
 BKGRCDDESC

 BKGPCDDATA
 BKGPCDDESC

 BKGULCDDATA
 BKGULCDDESC
}
```

The header data (project ID, etc.) not directly connected with a detector data window or an auxiliary data block are contained in the structure, shown below with typical values:

```
{HDR,
   ANGLE            DOUBLE         -0.20999999
   CATEGORY         STRING     'Science'
   CMP_NAME         STRING     ''
   CMP_NO           LONG                  0
   COMP_ERR         BYTE          0
   COMP_ID          LONG                  2
   COMP_OPT         LONG                  0
   DATE             STRING     '20/11/96'
   DATE_END         STRING     '1996-11-20T22:58:33.079Z'
   DATE_OBS         STRING     '1996-11-20T22:00:59.841Z'
   DETECTOR         STRING     'NIS'
   DW_ID            LONG                  5
   EV_ENAB          BYTE          0
   EXPCOUNT         LONG                 10
   EXPTIME          FLOAT           50.0000
   FILENAME         STRING     's5841r00.fits'
   GSET_ID          LONG                  0
   INSTRUME         STRING     'CDS'
   INS_ROLL         DOUBLE           0.0000000
   INS_X0           DOUBLE           399.00000
   INS_Y0           DOUBLE           54.400002
   IXWIDTH          FLOAT           20.3000
```

```
IYWIDTH         FLOAT          240.200
LL_ID           LONG                 0
MIR_POS         LONG               124
NWINDOWS        LONG                 6
NX              LONG                10
NY              LONG                 1
OBJECT          STRING    'QS'
OBJ_ID          STRING    ''
OBS_PROG        STRING    'NISAT_S'
OBT_END         DOUBLE       1.2272219e+09
OBT_TIME        DOUBLE       1.2272185e+09
OPSLBITS        STRING    '1011'
OPSRBITS        STRING    '1010'
OPS_L           LONG              1796
OPS_R           LONG              2189
ORIGIN          STRING    'SOHO-EOF'
PROG_NAME       STRING    '0'
PROG_ID         LONG                 0
PROG_NUM        LONG              5841
RAS_ID          LONG                16
RAS_VAR         LONG                 2
SCI_OBJ         STRING    'NIS QS Atlas'
SCI_SPEC        STRING    'NISAT_S/v2'
SEQ_VALID       BYTE         0
SEQ_IND         LONG                 0
SER_ID          LONG              5657
SFDUADID        STRING    'ZZZZ0001'
SLIT_NUM        LONG                 4
SLIT_POS        LONG                 0
STUDYVAR        LONG                 2
STUDY_ID        LONG                17
TELESCOP        STRING    'SOHO'
TITLE           STRING    'NIS SPECTRAL ATLAS (SPATIAL)'
TITLE_ID        STRING    '33'
TRACKING        BYTE         0
VDS_ACC         BYTE         0
VDS_MODE        LONG                 2
VDS_PMCP        LONG               100
VDS_ORI         BYTE         0
WAVEMAX         DOUBLE           632.87300
WAVEMIN         DOUBLE           307.56400
XCEN            DOUBLE           398.20001
```

```
   XSTEP           DOUBLE           2.0320001
   YCEN            DOUBLE           52.900002
   YSTEP           DOUBLE           0.0000000
}
```

The detector data windows are now stored by means of handles. Earlier, though, the data was accumulated into one contiguous array of the same number of dimensions as the detector data blocks. The first dimension (wavelength) was expanded to accommodate all the blocks in one array, `detdata`.

Information specific to each detector data window is stored in the array `detdesc`, with one element for each data window. These structures also contained the start and stop indices to be used when extracting detector data from the `detdata` array, in such a way that the actual data corresponding to detector data window index W was given by

```
DetData (DetDesc(W).ixstart(0):DetDesc.ixstop(0)     $ ;(for 1D Data)
         [ ,DetDesc(W).ixstart(1):DetDesc.ixstop(1)  $ ;(for 2D Data)
          [,DetDesc(W).ixstart(2):DetDesc.ixstop(2)]]$ ;(for 3D Data)
        )
```

It is not necessary for the general users to understand the exact functionality of the current data storage system, except that it's *highly* recommended to always use the `gt_xxx` routines to extract detector data from the QLDS. See CDS Software Note 41 for a description of these routines, or try `IDL>tftd,'gt_'` for a quick overview. Use e.g., `gt_windata()` to access the whole data block from one detector data (line)window.

The detector data descriptors contain the following tags:

```
{NISx/GISx      ; x == number of dimensions in the data set.

   HANDLE       ; An IDL handle pointing to the detector data array.

   IXSTART      ; Partially obsolete tags, but e.g.,
   IXSTOP       ; ABS(IXSTOP)-IXSTART+1 gives you an array with the
                ; sizes of the data block.  IF IXSTOP(0) IS NEGATIVE,
                ; THE DATA FROM THIS LINE WINDOW HAS NOT BEEN READ IN, AND
                ; IS THUS UNAVAILABLE}.

   PADDING      ; When data windows of differing sizes (detector y dimension),
                ; such as those with full spectrum NIS data are read in,
                ; they're padded out (with MISSING values) to a uniform size.
```

```
                  ; The original window size is given by
                  ; ABS(IXSTOP)-IXSTART+1 - PADDING

   LABEL          ; Name of the data extraction window.

   UNITS          ; The units of the data. Changes according to what
                  ; forms of calibrations have been applied to the data.

   MAX            ; The maximum/minimum value of the detector data
   MIN            ; from this line window.

   MISSING        ; The value of pixels with no data or corrupted data.

   AXES           ; String array with the names of the dimensions in the
                  ; data array, e.g., ['WAVELNTH','SOLAR_X','SOLAR_Y',
                  ; 'DEL_TIME']

   ORIGIN         ; The starting values (values for pixel (0,0,0,0) in the
                  ; data array) of the entities described by AXES (wavelength,
                  ; solar x, etc).
   SPACING        ; The spacing between values of the entities described by
                  ; AXES, i.e., wavelength for pixel (1,0,5,0) is
                  ; ORIGIN(0) + 1*SPACING(0), the solar y value is
                  ; ORIGIN(2) + 5*SPACING(2).
   ROTATION       ; Contains the rotation of each axis.
   DETX           ; The detector x value for pixel (0,0,..).
   BINX           ; Binning factor, detector x direction.
   DETY           ; Det. y value for pixel (0,0..)(Only present for NIS data)
   BINY           ; Binning factor, det. y direction (NIS only)
   WAVELENGTH     ; The central wavelength of the data window.
   WAVEMIN        ; The minimum wavelength of the data window.
   WAVEMAX        ; Maximum wavelength of the data window.
   WAVEBAND       ; Which wavelength band, 1-2 for NIS, 1-4 for GIS
   GR_ORDER       ; Grating order.
}
```

The auxiliary data are stored in a similar fashion, with one array for each type of data, and one descriptor structure to describe the accompanying information:

```
{AUXx,          ; x == number of dimensions

   LABEL        ; Name/type of the data
   UNITS        ;
   MISSING      ; The value used to represent missing data.
   AXES         ; Names of the dimensions in the auxiliary data.
   ORIGIN       ; See detector data descriptors.
   SPACING      ; See detector data descriptors.
   ROTATION     ; See detector data descriptors.
}
```

The presence of a certain type of auxiliary data can be checked with, for instance:

```
        if N_TAGS(qlds.mirpdesc) gt 0 then ; We have some Mirror position data.
```