CORONAL DIAGNOSTIC SPECTROMETER
SoHO

CDS SOFTWARE NOTE No. 25

Version 1.0 | Date 21 March 1995

COMMAND PREPARATION SOFTWARE DESIGN

Martin Carter
RAL

carter@cds4.bnsc.rl.ac.uk

1 General

1.1 Revisions

```
Version 0.0 : Draft for comment.
23/8/94

Version 0.1 : Changed use of PERL scripts. Now interact via files.
18/9/94

Version 0.2 : Removed DFDUMP, CONVERT_SERIES, PSDUMP perl scripts.
10/10/94      Changed intermediate file scheme.

Version 0.3 : Changed AUTO stuff.
13/10/94

Version 0.4 : Changed wording. Separated automatic mode tools into
01/11/94      separate document.

Version 0.5 : Added database descriptions.
02/11/94

Version 0.6 : Updated example files.
11/11/94

Version 0.7 : Added description of default macro tables.
15/02/95

Version 0.8 : Incorporated changes to cpt and datbases.
06/03/95

Version 0.9 : Made change to state structure passed to cpt.
17/03/95      Created new tables for appendix.

Version 1.0 : Changed TTC file example to reflect changes.
21/03/95

Version 1.1 : Updated table examples. Split tableID database into series\_id
30/5/95       and raster\_id databases. Added non time tagged study option
              back in. Added CDHS state database details.
              Modified definition of fundamental series.
```

1.2 Distribution

```
RAL  : MKC, CDP, RAH.
GSFC : WT, DZ.
MSSL : ERB.
```

1.3 Glossary

Reference [2] gives a definition of some of the terms used. The definitions of **bold** faced terms are repeated here.

- **Raster** : Fixed format 2-D scan whose parameters are stored in raster tables in the CDHS.
- **Series** : Set of commands executed consecutively and stored in the series tables in the CDHS.
- **Study** : Set of **rasters** with associated parameters or **series** and a time tag.
- **Science plan** : Set of **studies** for the day ahead.
- **System series** : **Series** reserved for system/engineering purposes.
- **Science series** : **Series** reserved for use by the **science plan** .
- **Automatic state** : The default state of CDS when running out of ground contact or autonomously.
- **Command preparation database** : database used and maintained by the command preparation tool.
- **Science telemetry monitor** : Program which intercepts the science telemetry stream and performs some initial on-line display.
- **Command preparation tool** : Tool for producing files need to implement the **science plan** .

1.4 Referenced Documents

1. CDS groundbased software requirements specification V2.0 29/3/93
2. CDS groundbased software design outline V1.0 2/4/93
3. CDS science planning tool conceptual design V1.1 (CDS SN7) 28/3/94
4. CDS planning CDP 14/4/94
5. The UIT Database System V1.1 6/6/94
6. CDS catalog definitions V0.24 (CDS SN2) 26/7/94
7. CDHS state database software V0.0 30/5/95

1.5 Summary and Purpose

This document decsribes the current implementation of the command preparation software. It arose from the command preparation software specification document. It details what tasks are performed by the command preparation software and how they are implemented.

2 Introduction

The **raster** forms the basic unit of observation in the CDHS. Any **study** consists of a time tagged sequence of **rasters** together with a few *glue* commands such as changing the pointing or switching on solar feature tracking.

Studies will usually have explicit start times associated with them but may also simply follow on consecutively from the previous **study** .

The technical planning tool [2,4] allows the user to design new **raster**s and to check their validity and provides information on the data product.

The science planning tool [3,4] allows the user to generate a **science plan** consisting of a number of time tagged **studies** and any other information needed.

Within the science planning and technical planning tools **studies** and **rasters** are defined by a number of parameters which are stored in databases [5] and may be referenced by a set of IDs. The **science plan** itself is also stored in a database and is referenced by time fields.

The command preparation software [4] accesses the databases and generates the commands to the CDHS needed to implement the **science plan** . This software assumes that the CDHS has been put into a standard state for autonomous operation. The standard PERL scripts and the CDHS state database maintain the definition of this state and send the relevant commands to the CDHS.

3 Command preparation software

The function of the command preparation software is to :

- Access the IDL databases and extract the study definitions forming the **science plan** .
- Produce the CDHS tables which are required by the **science plan** ;
- Provide tools for loading the tables onto the CDHS;
- Provide tools to verify that the tables necessary to run the **science plan** have been correctly loaded onto the CDHS;
- Maintain a database indicating what is currently loaded on the CDHS.

It is intended that the command preparation software minimize the number of commands sent to the CDHS by only uplinking tables which have changed. Only complete tables are loaded.

Because the contents of **series** tables may have a dependency on the start position of the study these may not be re-usable except in a few cases.

The command preparation software consists of two main tools :

- The **command preparation tool** (**cpt**), an IDL program which extracts the relevant **science plan** parameters from the databases and prepares the CDHS tables needed to implement it and maintains databases describing the tables currently loaded on the CDHS.
- EGSE/TCL perl scripts which load the **science plan** tables onto the CDHS and verify that the tables needed by the **science plan** have been correctly loaded onto the CDHS.

4 Command preparation tool

4.1 Introduction to the command preparation tool

The **science plan** is accessed by the **command preparation tool** . The **command preparation tool** is an IDL program which :

- prepares an ACSII file (the *.TTC file) containing the set of command mnemonics at specified times forming the **science plan** .
- maintains the **command preparation database** which contains the IDs and locations of the tables currently loaded on the CDHS;
- maintains the **series and raster ID database** which links the IDs of the CDHS **series** and **raster** tables with the planning database IDs defining the parameters used to construct the tables;
- recognizes which tables are required to be loaded onto the CDHS for this **science plan** ;
- generates the **series** tables need for the **science plan** containing the command mnemonics for running **rasters** and the extra *glue* commands needed. The tables are stored in ASCII files (series_index.CVT), overwriting previous versions;
- generates the raster, VDS window and data extraction window tables needed for the **science plan** and stores them in ASCII files labelled by the table type and CDHS index, overwriting previous versions (raster_index.CVT, vdswin_index.CVT and dexwin_index.CVT files);
- generates an ASCII file (*.LTB) containing the indeces of the series, raster, VDS window and data extraction window of all the tables that are used by the **science plan** . Any tables which need loading are identified.

4.2 Use of the command preparation tool

The **command preparation tool** takes the start and end times of the science plan period as inputs and extracts studies from the **science plan** database for this period. By default it works on a period starting at the start of the next UTC day and ending at the end of the same day.

The folowing options may be used with the **command preparation tool** :

- **startofperiod** : Override the start time of the science plan period. End time is end of same day.
- **endofperiod** : Override the end time of the science plan period.
- **now** : Define the start time of the science plan period to be the current utc time. End time is end of same day.
- **new** : Initialize the command preparation database (new=1,2) or the series and raster databases and delete all existing CVT, LTB and TTC files (new=2).
- **state** : Pass a structure describing the current state of the CDHS to the **command preparation tool** .

The **command preparation tool** is invoked from the IDL command line with the following :

```
IDL> cpt, startofperiod='1995-12-31 1:0:0', endofperiod='1996-1-1 4:0:0',
            state=state
```

The LTB and TTC files produced are labelled with the startofperiod time and have the form : '1995-12-31T01:00:00.000Z.TTC and LTB'.

4.3 Input from the science plan

The science planning tool produces a set of **studies** where each **study** consists of a number of consecutive **rasters** . The **studies** each have a time tag associated with them and a flag indicating whether the time tag should be used or whether the study should follow on consecutively from the previous one.

The parameter values used for each **study** and **raster** are stored in the planning databases maintained by the technical planning and science planning tools.

The **studies** are extracted from the **science plan** database by specifying the time interval required. This gives the study ID and and the study variation for each of the **studies** forming the **science plan** together with some comments on the nature of the study and some parameters defined at the **science plan** level.

These IDs are sufficient to extract the study details from the planning databases.

The extra parameters defined only at the **science plan** level are :

- start and end times of the study;
- flag indicating whether solar feature tracking is to be used;
- number of times last raster is to be run;
- an array giving a number of different initial pointings for the study where the study pointing is variable;

- the number of times the study is to be repeated at each pointing position;
- flag indicating whether the study is to be time tagged or follow consecutively.

4.4 Command preparation database

The command preparation database relates the CDS onboard table index to the internal CDHS ID for each of the loaded tables. This allows the **command preparation tool** to keep track of what tables are loaded where.

The database maintains information on series, raster, dexwin and vdswin tables. The information consists of :

- the last time the table was used;
- the tables indeces and table Ids of any tables loaded.

4.5 Series_id and raster_id databases

The series_id and raster_id databases relate the CDHS IDs used onboard the CDS to the IDs and parameters used on the ground for **series** and **raster** tables respectively.

It maintains a link between the IDs that are generated by the command preparation tool for each CDHS table entry and the set of database IDs that are used to define the parameters for the table. An entry in the database contains :

- the time the table entry was created;
- the set of science/technical planning database IDs that define the table;
- the ID assigned to the table by the command preparation software;

The command preparation tool only creates a new CDHS ID if a new set of planning database IDs is presented to it or a relevant parameter within one of the state databases has changed or if the table is not *fundamental* (the latter applies only to series tables, a non fundamental series has a non-time tagged study following it).

Some tables and IDs are initialized when the CDHS is powered on. These are not used by the **command preparation tool** .

4.6 CDHS state database

The CDHS state database [8] contains quasi-static parameters used to define the standard state for autonomous operation. They are referenced by time, command mnemonic and parameter number for a given mnemonic. Where parameters differ from their defaults or where the science or planning tools require the current setting for a parameter then the parameter is added to the database.

The quantities described within this database will eventually consist of things like :

- the current set of VDS, GIS, CDHS, EPS, MCU operating parameters;
- the version numbers of the command preparation software tools used to interpret the **science plan** ;
- the current set of pointing calculation parameters;
- the current set of solar feature tracking parameters;

Other quasi-static parameters, those not required by the planning or science tools and which do not fit neatly into the CDHS state database structure will be maintained elsewhere. These consist of things like :

- the current settings for various switches of the VDS, GIS, CDHS, EPS, MCU which are currently set up in mode tables;
- the parameters defining the current GIS look up tables;
- the current form of the CDHS mode tables;
- the current form of the CDHS **system series** tables;
- the current health monitor parameters;

5 Command preparation perl scripts

The command preparation perl scripts are run from the EGSE/TCL environment and access the command and telemetry databases. They use the TTC, LTB and CVT files produced by the **command preparation tool** .

The command preparation PERL scripts consist of :

- **DFLOAD** : a script to load the TTC file into the deferred command store in the CDHS.
- **RTLOAD** : a script to run through the TTC file in real time sending the commands to the CDHS;
- **PSLOAD** : a script which loads all the the new tables needed by the **science plan** onto the CDHS;
- **TBVAL** : a script to compare the contents of the tables dumped from the CDHS with the original tables to verify that they have been loaded correctly.

5.1 Load deferred commands PERL script (DFLOAD)

This PERL script translates the TTC file command mnemonics and times for the **science plan** produced by the **command preparation tool** into hex values and loads them into the deferred command store in the CDHS. It automatically dumps the contents of the deferred command store once loading is completed.

Note that, at the time of loading the next **science plan** into the deferred command store, part of the previous **science plan** will still be present.

The script is run from an EGSE/TCL session using :

```
run CPT/DFLOAD 1995-12-31T00:00:00.000Z.TTC
```

(The file extension may be omitted).

5.2 Load realtime commands PERL script (RTLOAD)

This PERL script translates the TTC file command mnemonics and times for the **science plan** produced by the **command preparation tool** into hex values and allows the commands to be sent to the CDHS in realtime commanding mode rather than from the deferred command store. It allows the user to intervene in the commanding and to see the results of the commands before proceeding.

The script runs through the contents of the TTC file prompting the user to send each command at the appropriate times and allowing the user to insert extra commands, skip commands or carry on.

The script is run from an EGSE/TCL session using :

```
run CPT/RTLOAD 1995-12-31T00:00:00.000Z.TTC
```

(The file extension may be omitted).

5.3 Load tables PERL script (PSLOAD)

This PERL script accesses the LTB file produced by the **command preparation tool** and identifies the new tables required for the **science plan** . It then loads the new tables required onto the CDHS. The script may be used off line to produce a BK file which can be loaded in background commanding mode. Any command mnemonics are first translated into hex values before loading.

The script then dumps all the tables required by the **science plan** as identified in the LTB file.

The tables to be loaded are produced by the **command preparation tool** and and are stored in *.CVT files labelled by the table type and index.

These file have the form :

```
series_index.CVT
raster_index.CVT
```

```
vdswin_index.CVT
dexwin_index.CVT
```

These files are also used by the command validation tool to check the tables have been loaded correctly.

The script is run from an EGSE/TCL session using :

```
run CPT/PSLOAD 1995-12-31T00:00:00.000Z.LTB
```

(The file extension may be omitted).

5.4 Science telemetry monitor

The contents of any special packets are extracted and placed into *.STM files by the **science telemetry monitor** . These are later compared with their expected contents by the command validation tool and the **automatic state** tools.

These file have the form :

```
series_index.STM
raster_index.STM
vdswin_index.STM
dexwin_index.STM
```

These files are have been loaded correctly.

The **science telemetry monitor** also extracts the contents of other non-science packets within the science stream and places them in files. These are :

- GIS raw data packets
- memory dump packets

5.5 The table validation PERL script (TBVAL)

This checks that the tables required by the **science plan** have been loaded correctly. These include :

- series tables;
- raster tables;

- data extraction windows tables;
- vds windows tables;
- deferred command store.

It is not currently envisaged that mode changes will be used by the science plan. Mode tables are not checked by TBVAL but must be maintained and checked by other means.

The PSLOAD perl script dumps all the relevant tables for the next **science plan** . The **science telemetry monitor** accesses the science stream and picks out the special packets containing the table data and places the data into files labelled by table type and table index with the file extension *.STM. These will be compared with the original tables stored in the *.CVT files used by PSLOAD and any discrepancies flagged.

The DFLOAD perl script dumps the contents of the deferred command store containing the commands to run the **science plan** . The **science telemetry monitor** places the contents of the deferred command store received in special telemetry into a file 'deferred.STM' for comparision with the original contents in the *.TTC file. Note that adjustment must be made for contents already in the deferred command store.

After the **science telemetry monitor** has received all relevant tables the TBVAL perl script is run to verify that the *.STM files agree with the *.CVT files and the *.TTC file. Any discrepancies are flagged.

5.6 Automatic state

The CDS is put into a standard well defined state : **automatic state** , before running the **science plan** so that unpredictable effects do not occur and to allow the time taken by each **study** to be accurately predicted.

The **automatic state** consists of :

- Deferred command store enabled;
- CDHS in READYBOTH state;
- default settings as described by the CDHS state database.

The **automatic state** is reached by running standard PERL scripts. The CDHS state database tools [8] are used to define other non-default parameter settings.

6 Operations

The EGSE/TCL software may be used to output commands in realtime to the CDHS or to generate BK files for background commanding mode commanding of the CDHS.

The realtime or background commanding mode used to uplink the **science plan** to the CDHS does not affect the **command preparation tool** so that the decision on whether to load the observation sequence into the deferred command store or to run through the commands in real time can be left to the EGSE/TCL stage. However, the TTC file should probably still be loaded into the deferred command store on the CDHS. The deferred command store would be disabled during the real time commanding period.

A typical sequence of events in operation might be :

1. Produce the **science plan** for the next autonomous period from the end of the current real time period to the end of the next real time period (typically a 24 hour period).
2. Run the command preparation tool on the **science plan** producing a *.TTC file; a *.LTB file and *.CVT files.
3. Run the PSLOAD perl script through the TCL tool to produce a command file which is then uploaded in background commanding mode .
4. Run the DFLOAD perl script through the TCL tool to produce a command file which is then uploaded in background commanding mode .
5. If real time control is required disable the deferred command store and assume real time control.
6. Once PSLOAD and DFLOAD have completed and the contents have been received by the **science telemetry monitor** run the TBVAL perl script.
7. Towards the end of the real time period put the CDHS into **automatic state** .

7 Appendix A : CDHS table formats

7.1 CDHS table types

The following tables are maintained by the MACRO module within the CDHS FM software.

Name	Number	Length (INT16s)	Description
RASTER	50	16	Raster parameter tables.
SERIES	60	128	Sequential command store.
DEXWIN	40	100	Data extraction windows.
VDSWIN	20	50	VDS windows.
MODE	10	50	Define the various CDHS modes.
IEFTAB	1	70	Inter-instrument flag detection parameters.

The other relevant table is the deferred command store. This is 1024 INT16 words long which corresponds to 256 normal length commands.

7.2 Default Macro tables

The following are a list of the default tables loaded on power up by the CDHS.

TYPE	Index	ID	Description
SERIES	0	0x0000	IMIF ON.
SERIES	1	0x0001	Heater table 1.
SERIES	2	0x0002	Heater table 2.
SERIES	3	0x0003	ESR sequence.
SERIES	10	0xFFFE	GIS default sequence.
SERIES	11	0xFFFF	VDS default sequence.
RASTER	0	0xFFFE	GIS default raster.
RASTER	1	0xFFFF	VDS default raster.
DEXWIN	0	0xFFFF	VDS default extraction window.
DEXWIN	1	0xFFFE	GIS default extraction window.
VDSWIN	0	0xFFFF	VDS default data window.

Mode tables are also loaded on power up and may be overwritten from the ground but the spare tables may not be written to so the table indeces are irrelevant.

On booting the CDHS : the IDs of the first 10 series tables are overwritten with 0-9; the default GIS and VDS sequences are copied to series tables 10 and 11; the default GIS AND VDS rasters and data extraction windows are copied to raster and dexwin tables 0 and 1; the default VDS window table is copied to vdswin table 0; on a cold start the first four default engineering sequences are copied to series tables 0-3. The above tables are therefore unavailable for use by the command preparation software.

8 Appendix B : Command preparation tool details

8.1 Internal CDHS IDs

The IDs used internally within the CDHS to reference the tables are generated for each new table. The VDSWIN and DEXWIN tables only depend on the planning database IDs and so the planning database IDs are used directly (though note that the VDS windows do depend on the VDS readout mode). The **series** and **raster** tables depend on the planning database IDs and variations as well as some of the parameters described at the science planning level. These extra parameters are also kept in the series_id and raster_id databases and a series or raster is only re-used if these parameters correspond as well as the ID and variation. For example, the initial position of each study is stored in the series_id database and the CDHS ID only re-used if a subsequent study has the same initial position as well as matching the other series parameters.

8.2 State structure variable

The **command preparation tool** may optionally be passed a structure variable containing information in the current state of the CDHS which may be used to affect the commands and parameters put in the series and raster tables.

```
Strcuture name : STATE_STRUCTURE
```

```
Tags :

  .detector : 'N','G','B' or 'X' single character indicating whether
                                 NIS, GIS, BOTH or NEITHER detector
                                 currently in READY mode.

  .opsl     : Current OPS left position.

  .ops      : Current OPS right position.

  .slitn    : Current slit number in use.
```

8.3 Command preparation tool run time output

The **command preparation tool** outputs information on each series, raster, dexwin or vdswin table loaded.

```
 CDHSID : the CDHS internal ID assigned to the table and passed back in
          telemetry.
 CDHSX  : the CDHS internal index indicating where the table is loaded
          on the CDHS.
 OK     : indicates whether table already loaded on CDHS.
```

9 Appendix C : Command preparation software file formats

All files are ASCII files and any text after a # sign is treated as a comment. The files are treated as a list of tokens. Each token is separated by blank spaces or line breaks. An item enclosed in quotes is treated as a single token. Parameters for scc commands should be given in the same token with blank separators. Parameters may be supplied in files with .dt file extensions. Script commands are interpreted. Times within TTC files must form a single token with blank, '/' or ':' separators.

9.1 Command preparation database

The **command preparation database** has the following definition file :

```
#title
Command preparation database

#maxentries
1000
```

```
#items
DATE        R*8   TAI time table last used
TYPE        C*6   String containing table type
CDHS_ID     I*2   Internal CDHS ID number
CDHS_X      I*2   Internal CDHS index

#index
CDHS_ID     sort/index
CDHS_X      sort/index
```

9.2 Series ID database

The series table ID database has the following definition file :

```
#title
Series ID database : Version 0.2 24/5/95

#maxentries
10000

#items
DATE        C*20  String containing UTC date entry created
SW_VERSION  C*4   String containing software version used to create table
CDHS_ID     I*2   Internal CDHS ID number
ID          I*2   Database ID number
VAR         I*2   Database variation number
FUND        C*1   Fundamental series
SFT         C*1   Solar feature tracking Y/N
SOLARX      I*2   Initial solar x coord of study
SOLARY      I*2   Initial solar y coord of study
SLITN       I*2   Initial slit number of study

#index
ID          sort/index
VAR         sort/index
```

9.3 Raster ID database

The raster table ID database has the following definition file :

```
#title
Raster ID database : Version 0.0 24/4/95

#maxentries
10000

#items
DATE        C*20  String containing UTC date entry created
SW_VERSION  C*4   String containing software version used to create table
CDHS_ID     I*2   Internal CDHS ID number
ID          I*2   Database ID number
VAR         I*2   Database variation number
REPN        I*2   Raster repeat number

#index
ID          sort/index
VAR         sort/index
```

9.4 Command preparation files

The following files are produced by the command preparation software.

Name	Directory	Contents
*.TTC file	CDS_CP_TTCFILES	Command mnemonics and times for **science plan** .
.LTB file	CDS_CP_LTBFILES	List of tables needed for **science plan** .
series_index.CVT	CDS_CP_CVTFILES	Series table command mnemonics.
raster_index.CVT	CDS_CP_CVTFILES	Raster table hex values.
vdswin_index.CVT	CDS_CP_CVTFILES	VDS window table hex values.
dexwin_index.CVT	CDS_CP_CVTFILES	Data extraction window table hex values.
series_index.STM	CDS_CP_STMFILES	Series table hex values.
raster_index.STM	CDS_CP_STMFILES	Raster table hex values.
vdswin_index.STM	CDS_CP_STMFILES	VDS window table hex values.
dexwin_index.STM	CDS_CP_STMFILES	Data extraction window table hex values.

9.5 Time tagged command (*.TTC) file

This is an ASCII file created by the command preparation tool which translates the **science plan** into a set of commands at particular times. It has the following format :

```
# DEFERRED TABLE FILE
# Created 1995/05/31 08:56:16.464
# Version 0.7
```

```
# Objective
# Time                              Command

# GSCT2 test plan
# study    repeat number = 0
# pointing repeat number = 0

#  tell mcu to move OPS left : 2048

"1995/05/18 00:00:00.000"   "CBMOPSLP 0x0800"

#  tell mcu to move OPS right : 2048

"1995/05/18 00:00:01.000"   "CBMOPSRP 0x0800"

#  tell mcu to move slit number : 4

"1995/05/18 00:00:02.000"   "CBMSLITN 0x0004"

#  run series command
#    CDHS series index = 12
#    CDHS series ID    = 12

"1995/05/18 00:00:16.000"   "CB5RUNS 0x000C"


# GSCT2 test plan
# study    repeat number = 0
# pointing repeat number = 0

#  tell mcu to move OPS left : 2048

"1995/05/18 01:27:30.000"   "CBMOPSLP 0x0800"

#  tell mcu to move OPS right : 2048

"1995/05/18 01:27:31.000"   "CBMOPSRP 0x0800"

#  tell mcu to move slit number : 4

"1995/05/18 01:27:32.000"   "CBMSLITN 0x0004"

#  run series command
#    CDHS series index = 12
#    CDHS series ID    = 12

"1995/05/18 01:27:33.000"   "CB5RUNS 0x000C"
```

This file is normally loaded into the deferred command store to implement the science plan for the next 24 hours. This is done using the PERL script file DFLOAD.

The TTC file may also be used for real time commanding of the CDHS using a perl script which prompts the user at the appropriate times (RTLOAD).

9.6 List of tables required (*.LTB) file

```
# LTB FILE
# Created 1995/05/31 08:56:15.843
# Version 0.2

# List of table indeces

# IDs:         12
NEWSERIES      12

# IDs:          2       3
NEWRASTER       2       3

# IDs:          9      11
NEWDEXWIN       2       3

# IDs:          9      11
NEWVDSWIN       1       2
```

9.7 Series table (series_index.CVT) file

The **series** tables may be formed either from hex values or command mnemonics. Note that the tables used for CB5FILS are suitable.

```
# SERIES TABLE FILE
# Created 1995/05/31 08:56:15.179
# Version 1.0
# Table = 1 of 1
# CDHS id 12
# CDHS index 12
# ID  = 0
# VAR = 1

# Fundamental study details
# study_id     = 0
# studyvar     = 1
# obs_prog     = UNNAMED
```

```
# title        = VDS GSCT2 test study (1.5 hours)
# category     = T
# n_rasters0   = 1
# n_rasters1   = 1
# sv_desc      = GSCT2 test 1.5 hours
# duration0    = 3525.00
# duration1    = 1725.00
# tracking     =

# Series ID : 12
0x000C

# Study ID / length : 0 / 46
0x002E

# increment study counter
"CB5INCS 0x0001"

# Fundamental rasters

# Raster 0

# Detector = N
#   Solar x   = 0 arcsecs
#   Solar y   = 0 arcsecs
#   no. of rasters   = 1

# tell macro to set VDS Exposure units
CB5SIC "CBVEXPTI 0x0004"

# tell macro mirror initial position : 128
"CB5MIR   0x0080"

# tell macro to tell mcu to move OPS left : 2048
CB5SIC "CBMOPSLP 0x0800"

# tell macro to wait : 100 msecs
"CB5WAIT  0x000A"

# tell macro to tell mcu to move OPS right : 2048
CB5SIC "CBMOPSRP 0x0800"

# tell macro to wait : 100 msecs
"CB5WAIT  0x000A"

# tell mcu to change slit number : 4
"CBMSLITN 0x0004"
```

```
# tell macro to wait : 100 msecs
"CB5WAIT  0x000A"

# run raster ID, index : 2
"CB5RUNR  0x0002"

# Raster 1

# Detector = N
#   Solar x   = 0 arcsecs
#   Solar y   = 0 arcsecs
#   no. of rasters   = 1

# tell macro to set VDS Exposure units
CB5SIC "CBVEXPTI 0x0004"

# tell macro mirror initial position : 128
"CB5MIR   0x0080"

# tell macro to tell mcu to move OPS left : 2048
CB5SIC "CBMOPSLP 0x0800"

# tell macro to wait : 100 msecs
"CB5WAIT  0x000A"

# tell macro to tell mcu to move OPS right : 2048
CB5SIC "CBMOPSRP 0x0800"

# tell macro to wait : 100 msecs
"CB5WAIT  0x000A"

# tell mcu to change slit number : 4
"CBMSLITN 0x0004"

# tell macro to wait : 100 msecs
"CB5WAIT  0x000A"

# run raster ID, index : 3
"CB5RUNR  0x0003"

# macro terminate sequence command
0x5F81 0xFFFF
```

9.8 Raster table (raster_index.CVT) file

The **raster** tables are formed from hex values. Note that the tables have the same form as those used for CB5FILR.

```
# RASTER TABLE FILE
# Created 1995/05/31 08:55:41.781
# Version 0.6
# CDHS id       2
# CDHS index       2
# RASTER ID  =        1
# RASTER VAR =        1

# detector  = N
# rasdesc   = (2x240) arcsec, (2x240) slit, (0,-) step size
# rv_desc   = (2x240) arcsec, 30s <QUIET>  Comp. scheme 1
# ll_desc   = GSCT#2 special
# dw_desc   = Background, All NIS1  4 lines.  dex = (30,120) pi

0x0002    # CDHS Raster ID
0x000C    # VDS table length :      12

# Compression scheme :       1
# Slit number        :       4

0x0140    # Compression scheme(byte), slitn(nibble), detector(nibble)

0x0001    # Number of runs of this raster :       1

# Number of mirror locations :      15
# Number of slit   locations :       1

0x0F01    # Number of mirror/slit locations

# Size of mirror steps :       0 arcsecs
# Size of slit   steps :       0 arcsecs

0x0000    # Size of mirror/slit steps

0x012C    # Exposure time :      30.0000 secs

0x0009    # CDHS Data extract window ID

0x0000    # Data compression option :       0


# VDS specific commands
```

```
0x0009    # CDHS VDS extraction window ID

# VDS orientation :       0

0x0200    # Orientation, CDHS map and mode, HTR status

0x0000    # Number of heater cycles

0x0000    # Number of heater cycles

0x0000    # VDS on-chip binning : off/on, xfac, yfac ( BYTE, NIBBLE, NIBBLE)
```

9.9 DEXWIN table (dexwin_index.CVT) file

The data extraction window tables are formed from hex values. Note that the tables have the same form as those used for CB5FILDE.

```
# DEXWIN TABLE FILE
# Created 1995/05/31 08:55:41.179
# Version 0.4
# CDHS id        9
# CDHS index        2
# ID  =        9

# detector  = N
# dw_desc   = Background, All NIS1  4 lines.  dex = (30,120) pi
# ll_desc   = GSCT#2 special

0x0009   # CDHS internal Dexwin ID
0x0024    # Table length

# window        0
# win_name WW_525_84

0x005D    # window xstart :      93
0x0200    # window ystart :     512
0x001E    # window xsize  :      30
0x0078    # window ysize  :     120

# window        1
# win_name WW_559_42

0x017C    # window xstart :     380
0x0200    # window ystart :     512
```

```
0x001E    # window xsize  :      30
0x0078    # window ysize  :     120

# window        2
# win_name WW_602_13

0x02E9    # window xstart :     745
0x0200    # window ystart :     512
0x001E    # window xsize  :      30
0x0078    # window ysize  :     120

# window        3
# win_name WW_606_69

0x0310    # window xstart :     784
0x0200    # window ystart :     512
0x001E    # window xsize  :      30
0x0078    # window ysize  :     120

# window        4
# win_name BB_NIS_1

0x0000    # window xstart :       0
0x0188    # window ystart :     392
0x0400    # window xsize  :    1024
0x0078    # window ysize  :     120

# window        5
# win_name Background

0x0032    # window xstart :      50
0x0032    # window ystart :      50
0x0032    # window xsize  :      50
0x0032    # window ysize  :      50

# window        6
# win_name Background

0x0320    # window xstart :     800
0x00C8    # window ystart :     200
0x0032    # window xsize  :      50
0x0032    # window ysize  :      50

# window        7
# win_name Background

0x0384    # window xstart :     900
0x0384    # window ystart :     900
```

```
0x0032    # window xsize  :      50
0x0032    # window ysize  :      50

# window        8
# win_name Background

0x0064    # window xstart :     100
0x0352    # window ystart :     850
0x0032    # window xsize  :      50
0x0032    # window ysize  :      50
```

9.10 VDSWIN table (vdswin_index.CVT) file

The VDS window tables are formed from hex values. Note that the tables have the same form as those used for CB5FILV.

```
# VDSWIN TABLE FILE
# Created 1995/05/31 08:55:40.628
# Version 0.3
# CDHS id        9
# CDHS index        1
# ID  =        9

# dw_desc   = Background, All NIS1  4 lines.  dex = (30,120) pi
# ll_desc   = GSCT#2 special

0x0009   # VDSwin ID
0x2212    # IMIF command header
0x3004    # VDS command header

# window        0

0x3400    # window xstart :       0
0x3988    # window ystart :     392
0x35FF    # window xstop  :     511
0x39FF    # window ystop  :     511

# window        1

0x3432    # window xstart :      50
0x3832    # window ystart :      50
0x347B    # window xstop  :     123
0x387B    # window ystop  :     123

# window        2
```

```
0x34AE    # window xstart :     174
0x38C8    # window ystart :     200
0x34DF    # window xstop  :     223
0x38F9    # window ystop  :     249

# window        3

0x3464    # window xstart :     100
0x387C    # window ystart :     124
0x3495    # window xstop  :     149
0x38AD    # window ystop  :     173

0x3C00    # VDS command terminator
```