EUV IMAGING SPECTROMETER
Hinode

EIS SOFTWARE NOTE No. 19

Version 2.1 21 June 2017

How to access and use the EIS databases

Peter Young
George Mason University
4400 University Drive
Fairfax, VA 22030
U.S.A.

pyoung9@gmu.edu

1 Overview

The EIS project makes use of the same IDL database structures and software that were used for the SOHO/CDS mission. These in turn were developed by Don Lindler for the HST/GHRS and IUE astronomy missions, and are referred to as the "UIT database system".

An important reference document on how the databases work and how they can be accessed is available as CDS Software Note #10, available from the CDS homepage (http://solar.bnsc.rl.ac.uk) and from the soho/cds branch of SSWDB. The reader is referred to this document for more details. The present document contains examples on how information about EIS studies can be obtained from the EIS databases.

A key feature of the UIT database system is that large amounts of information can be accessed very quickly, making it relatively easy to do complex searches on different parameters.

2 Summary of the EIS databases

There are several different EIS databases but this document will consider those databases that are most relevant to the general EIS user.

2.1 The as-run database

The two main databases are those that list the properties of the studies and rasters that were run by the instrument – the "as-run" database. (Note: there is an "as-planned" database for those studies that were scheduled to be run by the EIS Chief Observer, but not all of these studies will have actually been run.)

The as-run database actually consists of two distinct databases, called `eis_main` and `eis_experiment`. They are located in the directory

```
$SSW/hinode/eis/database/catalog
```

`eis_main` contains a list of all of the timeline entries, while `eis_experiment` contains a list of all the individual rasters. To illustrate the difference, consider how an EIS operation is planned. The EIS Chief Observer (CO) places a study on the EIS planning. The study can consist of multiple rasters, although usually it contains only one raster. The CO can set the study to have multiple raster repeats. Consider the following example:

Study	Raster 1	1 run
	Raster 2	10 runs

where the study consists of two rasters, the first raster is run once, while the second raster is run 10 times. The study has a unique timeline ID (`tl_id`) associated with it and will have a single entry in the `eis_main` database.

In the `eis_experiment` database, there will be 11 entries that each have the same `tl_id` value as the study in `eis_main`. These correspond to the 11 rasters of the study. Note that each of the 11 rasters will end up as a separate FITS file.

2.2 The raster and line-list databases

The two as-run databases do not contain all the information about the rasters that have been run, and so it may be necessary to extract information from the raster and line-list databases. The former contains information such as the raster step size, field-of-view and exposure time, while the latter contains the list of emission lines observed by the raster.

The raster (`eis_raster`) and line-list (`eis_linelist`) databases are stored in:

```
$SSW/hinode/eis/database/planning_db/technical_db
```

2.3 The as-planned databases

There are four databases related to the timeline, "as-planned" database, and they are stored in

```
$SSW/hinode/eis/database/planning_db/timeline_db
```

Basically these store the information for the plans that the EIS COs prepare. Note that not all of the planned studies will actually run. For example, a study may be aborted if EIS hits its telemetry limit.

Information about the rasters is stored in `eis_science2_db`. One difference with the `eis_experiment` database is that repeats of a raster will not have an entry. Consider the example from Sect. 2.1. In the as-planned database, this `tl_id` will only have two entries: one each for the two different rasters. Although the second raster was run 10 times, only the first of the rasters has an entry in the as-planned database.

3 Examples

3.1 Searching by time

The following gives an example of how the EIS catalog files can be searched. Suppose we want to find the raster that was running at 22:40 UT on 2007 January 20. The sequence of commands is as follows:

```
dbname=getenv('SSW')+'/hinode/eis/database/catalog/eis_experiment'
dbopen,dbname
t_obs='22:40 20-jan-2007'
t_tai=anytim2tai(t_obs)
t_tai_str=trim(t_tai,'(f15.3)')
query='date_obs < '+t_tai_str+', date_end > '+t_tai_str
list=dbfind(query)
dbext,list,'filename,tl_id',filename,tl_id
dbclose
print,filename
```

which will return the filename `eis_l0_20070120_223207.fits.gz`.

The reader should note the key parts of this query:

1. Open the database (dbopen).
2. Create a query string.
3. Perform the query (dbfind).
4. Extract information from the database (dbext).
5. Close the database (dbclose).

When determining what fields to extract from the database, the user should do:

```
dbhelp
```

(when the database is open) and a list of all of the fields will be displayed. The user can choose from any of these when making a query or extracting information.

3.2 Extracting study information

The eis_experiment database only contains information from the raster database. Suppose that we want to extract some of the meta-data from the study database that is associated with the raster found in the previous example.

The key database item that maps the raster database to the study database is the timeline ID number (TL_ID). In the previous example we extracted the timeline ID, which we find to be 697. We now open the study database to extract information for this timeline ID.

```
dbname=getenv('SSW')+'/hinode/eis/database/catalog/eis_main'
dbopen,dbname
query='tl_id = 697'
list=dbfind(query)
dbext,list,'stud_acr,obstitle,obs_dec',stud_acr,obstitle,obs_dec
dbclose
print,stud_acr,obstitle,obs_dec
```

You should find the following output:

```
PRY_loop_footpoints
Cool lines for study of loop footpoints (e.g., near sunspot); 80" r&
Cool lines for study of loop footpoints (e.g., near sunspot); 80" r&
```

The latter two outputs are fields intended to give information about the observation that are written by the EIS Chief Observer at the time the EIS plan is created.

3.3 Extracting detailed raster properties

Some properties of a raster are not stored in the `eis_experiment` catalog. Examples include exposure time(s) and the wavelength information for the data windows. These can be extracted from the raster and line list catalogs in the case of exposure times and wavelengths, respectively.

An example for how to extract the exposure times defined for a raster is given below. The key link to the `eis_experiment` catalog is the raster ID.

```
dbname=getenv('SSW')+'/hinode/eis/database/planning_db/technical_db/eis_raster_db
dbopen,dbname
query='id = 46'
list=dbfind(query)
dbext,list,'exposures',exposures
dbclose
print,exposures
```

The raster ID here belongs to the raster of the PRY_loop_footpoints study. The output `exposures` is an 8 element integer array that gives exposure times in milliseconds. The first entry is 30000 (30 s) and the rest are zeros. This indicates that there is only one exposure for this raster (a maximum of 8 are possible). The Solarsoft routine `eis_get_exposure_info` makes use of the raster catalog to extract a number of pieces of information relating to exposure time, and the reader is referred to this for more details.

For the line list catalog, each entry has an ID number that corresponds to the line list ID number (`ll_id`) in the raster catalog. For the raster considered above, one has:

```
dbname=getenv('SSW')+'/hinode/eis/database/planning_db/technical_db/eis_raster_db
dbopen,dbname
query='id = 46'
list=dbfind(query)
dbext,list,'ll_id',ll_id
dbclose
print,ll_id
```

We find the line list ID is 15, so we can now query the line list catalog to find the number of wavelength windows for the raster:

```
dbname=getenv('SSW')+'/hinode/eis/database/planning_db/technical_db/eis_linelist_db
dbopen,dbname
query='id = 17'
list=dbfind(query)
dbext,list,'n_lines',n_lines
dbclose
print,n_lines
```

The number of wavelength windows used by this raster is thus 20. Use the `dbhelp` routine to find the full list of items in the raster and line list catalogs.

4 The EIS_OBS_STRUCTURE routine

The routine `eis_obs_structure` (available in Solarsoft) serves as an example of how to extract information from the four databases described in this document. It is called as:

```
IDL> str=eis_obs_structure(t0,t1)
```

where ‘t0’ and ‘t1’ specify a time range. The routine identifies all rasters that were run during this time range and extracts meta-data, which goes into the output structure ’str’. The tags of this structure for one example are given below:

```
DATE_OBS   STRING    '2007-01-20T00:27:12.000'
DATE_END   STRING    '2007-01-20T00:31:40.000'
XCEN       FLOAT          175.542
YCEN       FLOAT          76.9085
FOVX       FLOAT          40.9344
FOVY       FLOAT          304.000
FILENAME   STRING    'eis_l0_20070120_002712.fits'
TL_ID      LONG              677
STUD_ACR   STRING    'PRY_fast_dens'
RAST_ACR   STRING    'Quick_AR_raster'
LINK       STRING    'http://tcrb.nrl.navy.mil/~iuu/eis/thumbnails/rasters/quick_ar_raster/eis'...
RAST_ID    INT            44
WAVELNTH   STRING    '19282 19512 20204 20383 25632'
WAVEMIN    STRING    '19246 19476 20177 20347 25587'
WAVEMAX    STRING    '19317 19547 20230 20418 25676'
OBSTITLE   STRING    'High cadence sit-and-stare in ARs; 5s exposure; Fe XIII dens; 40" c'
OBS_DEC    STRING    'High cadence sit-and-stare in ARs; 5s exposure; Fe XIII dens; 40" c'
HOP_ID     INT             0
SCI_OBJ    STRING    ''
SLIT_WIDTH INT             1
SIT_STARE  BYTE         0
STEP_SIZE  FLOAT          1.00000
SLIT_INDEX INT             0
NSTEPS     INT            40
NEXP       INT             1
EXP_TIMES  STRING    '5.0'
```

5 The uniqueness of TL_ID

Each *study* that is placed on the timeline is assigned a unique timeline ID, or TL_ID. As the mission progresses, the TL_ID numbers increase, although not necessarily monotonically as the EIS CO does not necessarily create the EIS plan by systematically adding studies from the beginning of the timeline onwards.

If you know the TL_ID, then you can retrieve information about the rasters by doing:

```
IDL> str=eis_obs_structure(tl_id=tl_id)
```

Note that there may be multiple entries as a single study can consist of multiple rasters.

There are three known problems with searching on the timeline ID, however, and these are discussed below.

5.1 Rasters before 23 November 2006

The EIS planning tool began being used from 13:14 UT on 23 November 2006. Prior to this EIS observations exist and were assigned TL_ID values that were generally quite high, but the planning tool does not recognise their existence. Therefore eventually as time passed, the planning tool reached these TL_ID values and reused them. An example is TL_ID=37200. Therefore if you search for one of these TL_ID values then you will find two entries widely separated in time.

For the `eis_obs_structure` routine, the pre-23 November studies are automatically filtered out, but you will see them if you use the UIT database software.

5.2 Response studies

Response studies are special cases that are assigned TL_ID values of 1, 3 and 4. If you search on one of these values, then you will find many entries widely spaced in time. Please check the website below for more details.

`http://solarb.mssl.ucl.ac.uk:8080/eiswiki/Wiki.jsp?page=TriggerStudies`

5.3 TL_ID values 16541 to 16603 (March 2010)

There is a period running from 07:33 on 20-Mar-2010 to 02:02 27-Mar-2010 for which the TL_ID values were duplicated. This is because the EIS CO who began planning at 11 UT 27-Mar-2010 did not have the most recent copy of the database, and so the planning tool began assigning TL_ID values that corresponded to the start of the previous week.

The consequence is that most of the TL_ID values between 16541 and 16603 are duplicated. Searching for one of these TL_ID values will thus lead to two entries, separated by about a week.

A Document modification history

Version 2.1: Added Sect. 2.3.
Version 2: Added Sect. 5.