

Example Use Cases From the SolarSoft HEK API

Introduction

A full API document is under development for the *Heliophysics Events Knowledgebase* (HEK). This document describes examples of the use of the existing SolarSoft interface to the HEK.

The SolarSoft HEK API

SolarSoft (SSW) provides IDL interface routines for both submitting event candidates for inclusion in the HEK and for querying the HEK for events matching a variety of constraints.

Querying the HEK Via IDL

For IDL queries to the HEK, SolarSoft provides two routines:

`ssw_her_query.pro` - Issues an input verbatim query to the HEK CGI interface and translates the output to an IDL structure vector of event matches.

`ssw_her_make_query.pro` - Generates a HEK CGI query string using style interface. for example, ssw times, and 'index' info are converted to HPKB <param>=<value> equivalents. output from this function provide examples of hpkb cgi API. Since it uses keyword inheritance without filtering (or checking), it accepts all current & future HPKB parameter pass through (but garbage in/garbage out) - current defaults are set to "most useful".

Examples

Example 1. Simple time range + event_type (flares here via /FL switch):

Step 1. Create the HEK query:

```
IDL> t0='1-jan-2000' & t1='1-jan-2001'
IDL> query = ssw_her_make_query(t0,t1,/fl) & more, query
event_starttime = 2000-01-01T00:00:00 &
event_endtime   = 2001-01-01T00:00:00 &
event_coordsys = helioprojective &
x1=-5000 & x2=5000 & y1=-5000 & y2=5000 & event_type=FL &
cosec=1 & type=column & cmd=search
```

Step 2. Execute the above HEK query:

```
IDL> events = ssw_her_query(query) &
help, events, events.fl
EVENTS          STRUCT = -> <Anonymous> Array[1]; Top level
<Expression>    STRUCT = -> <Anonymous> Array[100]
; per event_type structure vector
```

The 100 matches reflect the current default match limit - to change this and also illustrate keyword inheritance, change the call to `ssw_her_query.pro` by specifying `RESULT_LIMIT`:

```
IDL> query = ssw_her_make_query(t0,t1,/fl,result_limit=200)
&
more,query
event_starttime = 2003-01-01T00:00:00 &
event_endtime   = 2004-01-01T00:00:00 &
event_coordsys = helioprojective &
x1=-5000 & x2=5000 & y1=-5000 & y2=5000 & event_type=FL &
cosec=1 & type=column & cmd=search & result_limit=200
```

Example 2. Search queries may include one or more relational triplets of form:

{PARAMn=parameter, OPn=relation, VALUEn=value}

Relational operators include [=, >=, <=, <>, <, >, like]

These are realized via `SEARCH_ARRAY` keyword in `ssw_her_make_query`. For example, to search for SSW "latest events" flares \geq M5 for some time range:

```
IDL> query = ssw_her_make_query(t0,t1, $
search_array = ['fl_goescls>=M5', $
'frm_name=SSW Latest Events'],/fl) & more,query
event_starttime = 2003-01-01T00:00:00 &
event_endtime   = 2004-01-01T00:00:00 &
event_coordsys = helioprojective &
x1=-5000 & x2=5000 & y1=5000 & y2=5000 & event_type=FL &
cosec=1 & type=column & cmd=search & param0=fl_goescls &
op0=>= & value0=M5 & param1=frm_name &
op1== & value1=SSW Latest Events
```

Example 3. `ssw_her_make_query` accepts `ssw 'index'` (header records). In this case, times (optionally expanded +/- time window) and field of view are inherited and propagated to the associated cgi query - for example, using TRACE index records:

```
IDL> help,index & time_window,index & help,index,/str
INDEX          STRUCT      = -> <Anonymous> Array[20]
(first,last) = '1-DEC-03 00:29:53','1-DEC-03 19:54:53'
```

```

** Structure <21e3604>, 60 tags, length=1396, data
length=1390, refs=1:
  DATE          STRING      '03/12/01, 00:29:53.000'
  MJD           LONG        52974 ....
  ORIGIN        STRING      'TRACE'
  WAVE_LEN      STRING      '195'
..etc..

```

```

IDL> query=ssw_her_make_query(index, hour_window=[-2,1]) &
more, query
event_starttime=2003-11-30T22:29:53&event_endtime=2003-12-
01T20:54:53&event_coordsys=helioprojective&x1=732.331&x2=12
44.33&y1=423.660&y2=-
88.3401&event_type=*&cosec=1&type=column&cmd=search

```

```

IDL> event=ssw_her_query(query) & help,event,/str
** Structure <5b766244>, 2 tags, length=6912, data
length=6912, refs=1:
  CH            STRUCT      -> <Anonymous> Array[4]
  SS            STRUCT      -> <Anonymous> Array[2]

```

Example 4. Within SSW environment (ssw/hpkb application development for example), the routine struct4event.pro (ssw ontology branch) is available for determining current and valid parameter names; e.g., the structure tag names of event structure.

```

IDL> ar=struct4event('ar') & print, tag_names(ar.required)
IDL> EVENT_TYPE KB_ARCHIVDATE KB_ARCHIVID KB_ARCHIVIST
KB_ARCHIVURL EVENT_COORDSYS EVENT_COORDUNIT EVENT_ENDTIME
EVENT_STARTTIME EVENT_COORD1 EVENT_COORD2 (...etc...)

IDL> cme=struct4event('cme') &
print, tag_names(cme.optional)
EVENT_PROBABILITY EVENT_COORD3 EVENT_MAPURL EVENT_MASKURL
...
AR_NOAANUM AR_NUMSPOTS AR_POLARITY CME_ACCEL
CME_ACCELUNCERT CME_ACCELUNIT CME_MASS CME_MASSUNCERT (...etc...)

```

The full set of valid parameters and descriptions is available via:
http://www.lmsal.com/helio-informatics/hpkb/VOEvent_Spec.html