Quick Start Guide for the IDL CDAWEB_GET_BIN software package

The routine **cdaweb_get_bin** allows one to load data from NASA's cdaweb (http://cdaweb.gsfc.nasa.gov/) directly into your IDL session, a high speed internet connection is required. You don't have to manually download the data. The IDL software cdaweb_get_bin package can be downloaded at http://hpde.gsfc.nasa.gov/hpde_software_tools.html. Unzip the file cdaweb_get_bin.zip, and place the folder in a directory of your preference.

In order for IDL to locate the software the folder location must lie in your !PATH variable. The !PATH variable is string of ':' delimited directories that are searched when you run an IDL procedure or function (see http://www.exelisvis.com/docs/Managing_IDL_Paths.html).

Or, if you have the cdaweb_get_bin.sav file it must lie in a directory in given the !PATH variable.

NOTE:

If you also have installed SPEDAS (http://spedas.org/blog/), the SPEDAS directory must lie in !PATH after directory containing the cdaweb_get_bin source code. However if you use the .sav file instead of the source code the .sav file can lie anywhere in the path.

1. Example: Loading data into IDL using 'cdaweb_get_bin.pro'.

```
IDL> start_time = '1998-06-10T00:00:00.0Z' ;start time year-month-day of month, hour, minute second.
IDL> stop_time  = '1998-11-10T23:59:59.0Z' ;stop time
IDL> dt_sec=3600d0 ; time bin in seconds best to specify in double precison.
IDL> time_name='j_day' ; name of time variable, time is given in Julian days.
IDL> dataset_id = 'AC_H2_MFI' ;DATASET ID's
IDL> vars = ['Magnitude=B0', 'BGSEc=Bx,By,Bz'] ;CDAWEB variable names, on left of the = sign, user supplied on right.
IDL> cdaweb_get_bin,dataset_id,vars,start_time, stop_time,dt_sec,time_name=time_name
-------------------------------------------------------
CDF VARIABLE Magnitude, now called B0 whose independent variable is j_day
CREATED VARIABLE: B0[j_day]
Depend_0EPOCH
Fraction of missing binned points =     0.00784632
-------------------------------------------------------
CDF VARIABLE BGSEc, now called Bx,By,Bz whose independent variable is j_day , another dependency is cartesian
CREATED VARIABLE: Bx,By,Bz[cartesian, j_day]
Depend_0EPOCH
Fraction of missing binned points =     0.00784632
Fraction of missing binned points =     0.00784632
Fraction of missing binned points =     0.00784632
-------------------------------------------------------
TIME_NAME       STRING    = 'j_day'
T_OUT           DOUBLE    = Array[3696]
Created time variable j_day
Created dependency cartesian
Placed meta data in structure "cmbmeta" in calling level

IDL>
IDL> ;List of created variables
IDL>
IDL> help,j_day,B0,Bx,By,Bz,B0_nbin,Bx_nbin,By_nbin,Bz_nbin
J_DAY           DOUBLE    = Array[3696]
B0              FLOAT     = Array[3696]
BX              FLOAT     = Array[3696]
BY              FLOAT     = Array[3696]
BZ              FLOAT     = Array[3696]
B0_NBIN         FLOAT     = Array[3696]
BX_NBIN         FLOAT     = Array[3696]
BY_NBIN         FLOAT     = Array[3696]
BZ_NBIN         FLOAT     = Array[3696]
```

In this example the CDAW cdf variable names 'MAGNITUDE' and 'BGSEc' were renamed by using the equal sign. Because the cdf variable names are case sensitive the exact spelling of the variable name must be specified. The data is binned into uniform time bins whose size is specified by the variable 'dt_sec', its best to set dt_sec to double precission.

For this example B0, BX,BY and BZ are the average of the binned variables. The variable B0_NBIN, BX_NBIN,BY_NBIN and BZ_NBIN are the number of variables in each bin for the corresponding averaged variables B0, BX,BY and BZ. Bins with no data (i.e. B0_NBIN[j] is zero) are interpolated from adjacent bins with data.

If variable list vars = ['Magnitude=B0', 'BGSEc=Bx,By,Bz'] is replaced with vars = ['Magnitude', 'BGSEc'] then the variable with names MAGNETIUDE and BGSEC will be returned instead, where BGSEC (FLOAT = Array[3,3969]) is a vector.

The following commands: 'cmb_string_list, cmbnotes' and ' cmb_meta_list, cmbmeta ' can be used to keep tract of what you have loaded into IDL.

```
IDL> cmb_string_list, cmbnotes
      0
dataset_id=AC_H2_MFI input vars=Magnitude=B0, BGSEc=Bx,By,Bz, time_name=j_day
t_init=1998-06-10T00:00:00.0Zt_final=1998-11-10T23:59:59.0Zdt_sec=3600.0000
      1 B0[j_day]
      2 Bx,By,Bz[cartesian, j_day]
IDL> cmb_meta_list, cmbmeta
**********
AC_H2S_MFI
    **********
    B0
      fieldnam:B-field magnitude
      time dependenc:Epoch
      catdesc       :B-field magnitude
      var_notes     :
      units     :nT
    **********
    BX
      fieldnam:Mag Field vector, GSE coord
      time dependenc:Epoch
      catdesc       :Magnetic Field Vector in GSE Cartesian coordinates (1 hr)
      var_notes     :
      units     :nT
    **********
    ….
```

2. How to determine the 'dataset_id' and the 'cdf variable names'.

2a. A list of valid ids and variables names are given in the file 'List_of_valid_CDAWEB_dataset_IDs_and_variable_names.txt' included in the distribution. However the data posted on CDAWEB is dynamic and therefore the listing in this file is not complete.

2b. One can also determine valid ids and variable names at http://heliophysicsdata.gsfc.nasa.gov/websearch/dispatcher.

Note: the **dataset_id** is the same as the '**CDAS Product Id**'.

Say we want the dataset_id for the ACE Cosmic Ray Isotope Spectrometer

1. In box 'Text Restriction' in upper left, type in ACE.
   All datasets with ACE in its description will be displayed in the right column.
2. In the Cosmic Ray Isotope Spectrometer box click on button button “Get Data/Plots”
3. A new window is open in that window the data set ID is the CDAS Product Id
   a. So for the ACE Cosmic Ray Isotope Spectrometer its id is AC_H2_CRIS
   b. The column 'Parameter name' list the case sensitive variable names.

Note on this web page one can find the data set using various filters under 'Element Restriction'.

3. Keywords for cdaweb_get_bin are

KEYWORD	BRIEF DESCRIPTION
time_name	name of output time variable, default is 'TIMEJD'
autobad	if set, the data will be filtered for bad data points: data where |data - running mean| > sigmul*standard_deviation will be flagged as bad and not used.
sigmul	used only if autobad keyword is set: multiplicative factor of standard deviation for rejection of data: 5 (default), 4 (less agressive), 6 (more agressive).
tdas	if set, the variables are moved into TDAS, not written to the calling level.
to_struct	if set to a name the requested variables will be placed in a structure with that name on the calling level.
keepfiles	if set keep, the downloaded cdf files that contain the requested variable.
move_cdf_files_to_directory	if set to a directory and if keepfiles is set move the downloaded cdf files to that directory.
multiple_modes	If set to 0 (default) interpolate all modes to the dominant mode, if set to 1 the non-dominant modes are set to fill. This keyword was introduced because some multi-dimension variables are composed of multiple modes, for example ion flux as function of time and energy where energy is time dependent.

4. Potential conflicts between cdaweb_get_bin and your code.

All names in the distribution start with cdaweb_, cmb_, or spdf with the following exceptions tvread.pro, spike_editor.pro, creader_creator.pro and read_mycdf.pro.

The routine **spdfgetdata.pro** calls **read_mycdf.pro** which uses the following routines whose names could be in conflict with your code.

Below is a listing of code used my read_mycdf.pro:

Procedures:
ADD_MYCOMPONENTS
ADD_MYDELTAS
ADD_MYDEPENDS
CALDAT
COMPUTE_CRDS
DRTOLLP
DUNIT
FOLLOW_MYDEPENDS
GEIGEO
GEIGSE
GET_ROTM
GET_SCALARP
ICAL
PTG
PTG_NEW
RECALC
SUN
UVILOOK
VECTOR_TO_RA_DEC
XV_LOOKV_TO_GCI

Functions:
ADD_SECONDS
ALTERNATE_VIEW
AMI_ISTPPTR
AMI_VAR
APPEND_MYDATA

APPLY_ESA_QFLAG
APPLY_QFLAG
APPLY_RTN_CADENCE
APPLY_RTN_QFLAG
ATAN2D
BREAK_MYSTRING
BUF_TRAP
CALC_P
CDFTYPE_TO_MYIDLTYPE
CHECK_MYVARTYPE
CLEAN_DATA
COMPUTE_CADENCE
COMPUTE_MAGNITUDE
COMP_THEMIS_EPOCH
CONVERT_LOG10
CONVERT_NI
CONVERT_TOEV
CONV_MAP_IMAGE
CONV_POS
CONV_POS_HUNGARIAN
CORRECT_FAST_BY
CORRECT_VARNAME
CORRECT_VNAMES
COSD
CREATE_PLAIN_VIS
CREATE_PLMAP_VIS
CREATE_VIS
CROP_IMAGE
DATAND
DFMAG
DTAND
ENCODE_CDFEPOCH
ERROR_BAR_ARRAY
EXPAND_WAVE_DATA
EXTRACT_ARRAY

FGEODEP
FIND_EPOCHVAR
FIND_VAR
FLIP_IMAGE
GETVAR_ATTRIBUTE_NAMES
GET_ALLVARNAMES
GET_MYDATA
GET_NUMALLVARS
HEIGHT_ISIS
HSAVE_STRUCT
MAKE_STACK_ARRAY
MERGE_METADATA
MODIFY_MYSTRUCT
PARSE_DISPLAY_TYPE
PARSE_MYDEPEND0
READ_MYATTRIBUTE
READ_MYCDF
READ_MYMETADATA
READ_MYVARIABLE
REGION_FILT
REPCHR
REPLACE_BAD_CHARS
SHIFTDATA_ABOVE_ONE
SHIFTDATA_ABOVE_ZERO
SIND
TAGINDEX
TIMESLICE_MYSTRUCT
UNIQ
UNIQUE_ARRAY
VTYPE_NAMES
VV_NAMES
WIND_PLOT
WRITE_FILL

5. cdaweb_get_bin -header

```
;
;Copyright 1996-2013 United States Government as represented by the
;Administrator of the National Aeronautics and Space Administration.
;All Rights Reserved.
; This software may be used, copied, or redistributed as long as it is not
; sold and this copyright notice is reproduced on each copy made.  This
; routine is provided as is without any express or implied warranties
; whatsoever.
;

pro cdaweb_get_bin,dataset_id,vars, t_init, t_final,dt_sec,time_name=time_name, autobad=autobad,
sigmul=sigmul,$
                tdas=tdas,
to_struct=to_struct,keepfiles=keepfiles,move_cdf_files_to_directory=move_cdf_files_to_directory, $
                multiple_modes=multiple_modes

;+
; EXAMPLE USAGE:
;   cdaweb_get_bin, 'AC_H3_MFI', ['Magnitude', 'BRTN'], '2007-01-10T02:00:00.0Z',
'2007-01-10T23:59:59.0Z',time_name='Julian_day'
;
; NAME:
;   cdaweb_get_bin
;
; PURPOSE:
; This procedure retrieves data remotely from the SPDF/CDAWeb and
; creates uniformly spaced binned data for the specified variables and data set.
; Bins containing no data are interpolated using nearest neighbor bins that contain data.
; It then creates these variables and their metadata on the calling level of your current IDL session.
;
; CATEGORY:
; Remote data retrieval and binning.
;
; CALLING SEQUENCE:
; cdaweb_get_bin,dataset_id,vars, t_init, t_final,dt_sec,time_name=time_name, autobad=autobad,
sigmul=sigmul,$
;       tdas=tdas,
to_struct=to_struct,keepfiles=keepfiles,move_cdf_files_to_directory=move_cdf_files_to_directory
;
; INPUTS:
;   dataset_id: cdf dataset id   i.e. dataset_id = 'AC_H3_MFI'
;       Note: the dataset_id and cdf variable names can be determined at
http://heliophysicsdata.gsfc.nasa.gov/websearch/dispatcher.
;   vars: list of cdf variable names,  i.e. vars = ['Magnitude', 'BRTN']
;       Note: cdf variable names are case sensitive.
;       The default output variable name is the cdf variable name.
;       To rename output variables and/or break them into components use following syntaxt:
;       vars =['cdfvariablename1=var','cdfvariablename2=var1,var2,var3']
```

; the later breaks 'cdfvariablename2' into components named 'var1','var2','var3',
; Note: the number of specified output components must equal the number of components for that cdf variabe.
; For example vars = ['Magnitude=B0', 'BRTN=Bvec'],
; would create output variables named 'B0' and 'BVEC'.
; For example vars = ['Magnitude=B0', 'BRTN=Bx,By,Bz'],
; would create output variables named 'B0' and 'Bx','By','Bz' (components) instead of 'Magnitude' and 'BRTN' (vector).
; t_init: start time in isodate format, i.e. t_init ='2007-01-10T02:00:00.0Z'
; t_final: stop time in isodate format, i.e. t_final='2007-01-10T23:59:59.0Z'
; dt_sec: time interval in seconds of the time bin width.
;
; Keyword Inputs:
; time_name: name of output time variable, default is 'TIMEJD'.
; autobad - if set data will be filtered for bad data points: data where |data - running mean| > sigmul*standard_deviation
; will be flagged as bad and not used.
; sigmul - used only if autobad keyword is set: multiplicative factor of standard deviation for rejection of data: 5 (default), 4 (less agressive), 6 (more agressive).
; tdas - if set the variables are moved into TDAS, not written to the calling level.
; keepfile - if set keep the downloaded cdf files that contain the requested variable.
; move_cdf_files_to_directory - if set to a directory and if keepfiles is set move the downloaded cdf files to that directory.
; to_struct - if set to a name the requested variables will be placed in a structure with that name on the calling level.
; multiple_modes =0 (default) interpolate all modes to the dominant mode, =1 set the non-dominant modes to fill, this keyword was introduced
; because for some multi depend variable are composed of multiple modes, like ion flux as function of time and energy where energy is time dependent.
;
; OUTPUTS:
; <variable>: binned data whose name is specified by input variable 'vars'.
; <variable>_NBIN; number of data points in a given bin.
; time_name: center time in Julian Days of each data bin.
; Note: these variables will be created on the calling level.
;
; COMMON BLOCKS:
; None.
;
; SIDE EFFECTS:
; Unknown.
;
; RESTRICTIONS:
; Unknown.
;
; PROCEDURE:
;
; MODIFICATION HISTORY:
; Code developed by Aaron Roberts and Scott Boardsen at GSFC.

;-

6. spike_editor -header

```
;
;Copyright 1996-2013 United States Government as represented by the
;Administrator of the National Aeronautics and Space Administration.
;All Rights Reserved.
; This software may be used, copied, or redistributed as long as it is not
; sold and this copyright notice is reproduced on each copy made.  This
; routine is provided as is without any express or implied warranties
; whatsoever.
;

function spike_editor,datain,sigmul,fillval=fillval,nsum=nsum,igood=ig
;+
; NAME:
;   spike_editor
;
; EXAMPLE USAGE:
;   given time series y of x, filter bad data points in y
;   y=spike_editor(y,sigmul,fillval=-1e31,igood=ig)
;   plot,x[jg],y
; PURPOSE:
; To filter data returning data points judged to be good.
; Based on D. A. Roberts, "An algorithm for finding spurious points in turbulent signals.",
; COMPUTERS IN PHYSICS, JOURNAL SECTION, SEPT/OCT 1993.
;
; CATEGORY:
; Data fitering.
;
; CALLING SEQUENCE:
; result= cmb_autobad(datain,sigmul,fillval=fillval,nsum=nsum)
;
; INPUTS:
;   datain - one or two dimensional data array of n n_elements: datain[2,n] or datain[n].
;   sigmul - multiplicative factor of standard devidation in each sub array: 5 (default).
;          Absolute values of the (data - mean) > sigmul*standard_deviation
;          will be flagged as bad (0), if less than then flagged as good (1).
; Keyword Inputs:
;   fillval - fill values: -1e31 (default).
;   nsum - size of sub-array for each filter step: 100 (default).
;
; OUTPUTS:
;  filtered data
; Keyword Outputs:
;  igood - indices of input data array datain judged to be valid data points
; COMMON BLOCKS:
;   None.
```

;
; SIDE EFFECTS:
;   Unknown.
;
; RESTRICTIONS:
;   Unknown.
;
; PROCEDURE:
;
; MODIFICATION HISTORY:
;   Code developed by Aaron Roberts and Scott Boardsen at GSFC.
;-

7. creader_creator

This routine allows you to create creaders.

Note if not compiling directly from the source code, you must
IDL>restore,'cdaweb_get_bin.sav'

The file 'List_of_valid_CDAWEB_dataset_IDs_and_variable_names.txt' must lie in the current IDL directory. This file contains the
data set ids and variable names and their descriptions.

```
IDL> creader_creator
% Compiled module: FILE_WHICH.
% Compiled module: CMB_FILE_EXISTS.
**************** LIST of OPTIONS ********************
      0 List dataset ids with brief description
      1 Input dataset id filter
      2 Input keyword filter
      3 Select dataset id
      4 Create creader
      5 Reset id list
      6 Quit program
Input # of option:
```

For example to create an RBSP-A magnetometer creader.

1. Input option 2
2. Input keyword 'magnetometer'
3. A list of all datasets with magnetometer in their description will appear.
4. Input option 3 to select the dataset you want.
   Here RBSP-A 1 sec data was chosen.
5. Input option 4 to create a creader

```
Input # of option:4
input the width of the time bin dt_sec in seconds (float or double):1d0
input name of creader you are creating (e.g. cread_test.pro or full path like
/home/user/cread_test.pro):cread_rbsp_mag_1sec.pro
```

For this example we create a creader call cread_rbsp_mag_1sec.pro.

Edit the start/stop times in cread_rbsp_mag_1sec.pro, or variable list.

```
timename='j_day'
dt_sec=1d0
start_time = '2014-08-30T00:00:05.000Z'
stop_time = '2014-08-31T23:59:53.000Z'
dataset_id = 'RBSP-A_MAGNETOMETER_1SEC-GEI_EMFISIS-L3'
; Fluxgate magnetometer data - Craig Kletzing (University of Iowa)
vars = [ 'Mag',  'Magnitude',  'delta',  'lambda',  'rms',  'coordinates',  'range_flag',  'partition',  'MET',
'calState',  'magInvalid',  'magFill']
vars = [ 'Mag',  'Magnitude',   'coordinates' ]
; d = spdfgetdata(dataset_id, vars, [start_time, stop_time])
cdaweb_get_bin,dataset_id,vars,start_time, stop_time,dt_sec,time_name=timename
end
```

then

idl>.run cread_rbsp_mag_1sec.pro

to load the data.

```
IDL> cmb_meta_list,cmbmeta
************************
RBSP_A_MAGNETOMETERS_1SEC_GEI_EMFISIS_L3
    ************************
    MAG
      fieldnam:Mag
      time dependenc:Epoch
      catdesc       :Magnetometer vector
      var_notes     :
      units     :nT
    ************************
    MAGNITUDE
      fieldnam:Magnitude
      time dependenc:Epoch
      catdesc       :Average magnitude (average of |B| values in the interval)
      var_notes     :
      units     :nT
    ************************
    COORDINATES
      fieldnam:Mag
      time dependenc:Epoch
      catdesc       :S/C position vector
      var_notes     :
      units     :km
```