afs.pl

(Auto Form Submit, ACE File Server, Anemic Fetch Slave, ???)

A Perl Script Providing Remote Access to the ACE Science Data Archives)

Syntax

`afs.pl` *`args`*

`or`

perl afs.pl *`args`*

Overview

The Perl script **afs.pl** provides the capability to retrieve data from the Ace Science Center (ASC) data archives via the Web, without using a web browser. Data retrieval from the ASC was previously only via a series of interactive web screens.

afs.pl was written to test the existing (legacy) Applications Programming Interface (API) to the ACE data via the internet. It may in the future become a component of a middleware that would connect the ASC to future virtual observatories. It may also be useful as a stand-alone application. No changes to the existing ACE web data server API were made to accommodate afs.pl. **afs.pl** simply makes use of features of the ACE web data server that have existed for years. Changes to the API may be made in the future to better accommodate programs like **afs.pl**.

The capability to support requests for simple plots of the ACE data has recently been added. Additional optional arguments are available for some control of plot appearance. (see the Plots section below)

The *afs.pl* program is called by invoking the program name with parameter strings on the command line. Output is to stdout.

Assumptions and Limitations

Only some basic error-checking of the command line arguments is performed.

MAG, SWEPAM, EPAM, and SWICS Level 2 data may be requested via afs.pl at this time. Other ACE instruments may be added at a later date.

Dates bounding data requests of MAG 16sec and 4min averages, SWEPAM 64sec averages, EPAM 5-minute averages, and SWICS hourly averages, may NOT span year boundaries. Dates bounding data requests of hourly averages (except for swics) and daily averages may span year boundaries.

For requests that generate large byte-count replies, the data are zip-compressed, and an HTML message containing the URL of the compressed file is returned instead of the data itself. Example: MAG 16sec averages for a 200 day period. This is an example of a case where the existing ACE data server API could be improved.

It is assumed that the Perl, and the LWP Perl library making this whole thing work, is installed on the users computer. See the libwww-perl section below for details.

Commandline Arguments

A list of space-separated strings.

The order of the arguments is not important, except for arguments following the –s and –e and the optional –title and –ytitle args. See below.

Upper/Lower case of letters within arguments is not significant.

Instrument arguments

One of the following instrument args is required:

`mag` `swepam` `epam` `swics`

(not yet implemented: `cris` `sepica` `sis` `uleis`

Time Average arguments

For MAG, one of the following time average args is required:

```
16sec     4min      1hr       1day
```

For SWEPAM, one of the following time average args is required:

```
64sec               1hr       1day
```

For SWICS, one of the following time average args is required:

```
                    1hr       2hr
```

For EPAM, one of the following time average args is required:

```
5min                1hr       1day
```

Note: “hourly” may be substituted on the commandline in place of “1hr”. Similarly, “daily” may be used instead of “1day”.

Start and End Date arguments

A start date and an end date are required:

-s *YR_start DOY_start* example: `"-s 1999 3"`

-e *YR_end DOY_end* example: `"-e 1999 22"`

Note: Year values may be 2-digit or 4-digit values. Day of year values may be 1, 2 or 3 digit values with leading zeros OK.

Data arguments

One or more data args are required:

For MAG:

```
year, day, hr, min, sec: year, day of year, hour of day, minutes, seconds.
fp_year       : fractional year.
```

```
fp_doy        : fractional day-of-year.
ACEepoch      : seconds since Jan 1 00:00:00 UT 1996.
SCclock       : ACE spacecraft clock counter (seconds).
Brtn_r        : R-component of mag. field in RTN (nT).
Brtn_t        : T-component of mag. field in RTN (nT).
Brtn_n        : N-component of mag. field in RTN (nT).
Bmag          : <|B|> magnetic field magnitude (nT).
Delta         : RTN latitude.
Lambda        : RTN longitude.
Bgse_x        : X-component of mag. field in GSE (nT).
Bgse_y        : Y-component of mag. field in GSE (nT).
Bgse_z        : Z-component of mag. field in GSE (nT).
Bgsm_x        : X-component of mag. field in GSM (nT).
Bgsm_y        : Y-component of mag. field in GSM (nT).
Bgsm_z        : Z-component of mag. field in GSM (nT).
dBrms         : RMS values of underlying high-resolution measurements (nT).
sigma_B       : Variance of |B|, i.e. sqrt(<(|B|-<|B|>)^2>) (nT).
fraction_good : Fraction of the period for which data was available
N_vectors     : Num of hi-res (16-sec) points included in the average.
Quality       : 0 = Normal data.
              : 1 = Spacecraft Maneuver and high-nutation period

              : 2 = Bad or missing data, (data values set to -999.9)
pos_gse_x
pos_gse_y
pos_gse_z     : Components of spacecraft position in GSE (km).
pos_gsm_x
pos_gse_y
pos_gse_z     : Components of spacecraft position in GSM (km).
```

For SWEPAM:

```
year, day, hr, min, sec: year, day of year, hour of day, minutes, seconds.
fp_year       : fractional year.
fp_doy        : fractional day-of-year.
ACEepoch      : seconds since Jan 1 00:00:00 UT 1996.
H_density     : Proton Density (cm^-3).
H__temp       : Radial Component of proton temperature (deg. Kelvin).
Alpha_ratio   : Ratio of alphas/protons
H_speed       : Proton Speed (km/s)
Vgse_x
Vgse_y
Vgse_z        : X,Y,Z-components of proton velocity in GSE coords (km/s)
Vrtn_r
Vrtn_t
Vrtn_n        : R,T,N-components of proton velocity in RTN coords (km/s)
Vgsm_x
Vgsm_y
Vgsm_z        : X,Y,Z-components of proton velocity in GSM coords (km/s)
pos_gse_x
pos_gse_y
pos_gse_z     : Components of spacecraft position in GSE (km).
pos_gsm_x
```

```
pos_gse_y
pos_gse_z     : Components of spacecraft position in GSM (km).
```

For SWICS:

```
year,day,hr,min,sec: year, day of year, hour of day, minutes, seconds.
fp_year     : fractional year.
fp_doy      : fractional day-of-year.
ACEepoch    : seconds since Jan 1 00:00:00 UT 1996.
SCclock     : ACE spacecraft clock counter (seconds).
nHe2        : He++ density (1/cm^3)
vHe2        : He++ velocity (km/s)
vthHe2      : He++ Thermal velocity (km/s)
vC5         : C+5 velocity (km/s)
vthC5       : C+5 Thermal velocity (km/s)
vO6         : O+6 velocity (km/s)
vthO6       : O+6 Thermal velocity (km/s)
vFe10       : Fe+10 velocity (km/s)
vthFe10     : Fe+10 Thermal velocity (km/s)
C6to5       : C+6/C+5 Charge state ratio
O7to6       : O+7/O+6 Charge state ratio
avqC        : C Average Charge state
avq_O       : O Average Charge state
avq_Fe      : Fe Average Charge state
FetoO       : Fe/O Element Ratio
HetoO       : He/O Element Ratio
CtoO        : C/O Element Ratio
NetoO       : Ne/O Element Ratio
MgtoO       : Mg/O Element Ratio
SitoO       : Si/O Element Ratio
XXX_qual    : Item XXX Quality Flag: 0->good data. Non-zero->see release
notes
```

For EPAM:

```
year,day,hr,min,sec: year, day of year, hour of day, minutes, seconds.
fp_year: fractional year.
fp_doy:  fractional day-of-year.
ACEepoch:seconds since Jan 1 00:00:00 UT 1996.
W3:      CA60 ,(0.389-1.28 MeV/nuc. He),Sector Avg,1/(cm**2-s-sr-MeV/nuc.).
W4:      CA60 ,(1.28-6.98 MeV/nuc. He),Sector Avg,1/(cm**2-s-sr-MeV/nuc.).
W5:      CA60 ,(0.465-1.71 MeV/nuc. CNO),Sector Avg,1/(cm**2-s-sr-
MeV/nuc.).
W6:      CA60 ,(1.71-19.1 MeV/nuc. CNO),Sector Avg,1/(cm**2-s-sr-MeV/nuc.).
W7:      CA60 ,(0.239-0.840 MeV/nuc. Fe (9&#60;Z&#60;29)),Sector
Avg,1/(cm**2-s-sr-MeV/nuc.).
W8:      CA60 ,(0.840-92.7 MeV/nuc. Fe (9&#60;Z&#60;29)),Sector
Avg,1/(cm**2-s-sr-MeV/nuc.).
Z2:      WARTD ,(Z>1,E>0.7 MeV Ions),Sector Avg,1/(cm**2-s-sr).
Z2A:     WARTD ,(Z>7,E>7.5 MeV Ions),Sector Avg,1/(cm**2-s-sr).
Z3:      WARTD ,(Z>5,E>2.5 MeV Ions),Sector Avg,1/(cm**2-s-sr).
```

```
Z4:      WARTD ,(Z>10,E>9.0 MeV Ions),Sector Avg,1/(cm**2-s-sr).
DE1:     DE ,(0.038-0.053 MeV Electrons),Sector Avg,1/(cm**2-s-sr-MeV).
DE2:     DE ,(0.053-0.103 MeV Electrons),Sector Avg,1/(cm**2-s-sr-MeV).
DE3:     DE ,(0.103-0.175 MeV Electrons),Sector Avg,1/(cm**2-s-sr-MeV).
DE4:     DE ,(0.175-0.315 MeV Electrons),Sector Avg,1/(cm**2-s-sr-MeV).
E1:      LEFS150 ,(0.045-0.062 MeV Electrons (+Ions) ),Sector Avg,1/(cm**2-
s-sr-MeV).
E2:      LEFS150 ,(0.062-0.102 MeV Electrons (+Ions) ),Sector Avg,1/(cm**2-
s-sr-MeV).
E3:      LEFS150 ,(0.102-0.175 MeV Electrons (+Ions) ),Sector Avg,1/(cm**2-
s-sr-MeV).
E4:      LEFS150 ,(0.175-0.312 MeV Electrons (+Ions) ),Sector Avg,1/(cm**2-
s-sr-MeV).
FP5:     LEFS150 ,(0.540-0.765 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
FP6:     LEFS150 ,(0.765-1.22 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
FP7:     LEFS150 ,(1.22-4.94 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P1:      LEMS30 ,(0.047-0.065 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P2:      LEMS30 ,(0.065-0.112 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P3:      LEMS30 ,(0.112-0.187 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P4:      LEMS30 ,(0.187-0.310 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P5:      LEMS30 ,(0.310-0.580 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P6:      LEMS30 ,(0.580-1.06 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P7:      LEMS30 ,(1.06-1.91 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P8:      LEMS30 ,(1.91-4.75 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
E1p:     LEFS60 ,(0.045-0.062 MeV Electrons (+Ions) ),Sector Avg,1/(cm**2-
s-sr-MeV).
E2p:     LEFS60 ,(0.062-0.103 MeV Electrons (+Ions) ),Sector Avg,1/(cm**2-
s-sr-MeV).
E3p:     LEFS60 ,(0.103-0.175 MeV Electrons (+Ions) ),Sector Avg,1/(cm**2-
s-sr-MeV).
E4p:     LEFS60 ,(0.175-0.312 MeV Electrons (+Ions) ),Sector Avg,1/(cm**2-
s-sr-MeV).
FP5p:    LEFS60 ,(0.546-0.761 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
FP6p:    LEFS60 ,(0.761-1.22 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
FP7p:    LEFS60 ,(1.22-4.97 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P1p:     LEMS120 ,(0.047-0.066 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P2p:     LEMS120 ,(0.066-0.114 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P3p:     LEMS120 ,(0.114-0.190 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P4p:     LEMS120 ,(0.190-0.310 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P5p:     LEMS120 ,(0.310-0.580 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P6p:     LEMS120 ,(0.580-1.05 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P7p:     LEMS120 ,(1.05-1.89 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
P8p:     LEMS120 ,(1.89-4.75 MeV Ions),Sector Avg,1/(cm**2-s-sr-MeV).
unc_XX :  Fractional uncertainty (statistical) in the indicated flux.
```

Example

perl `afs.pl mag 16sec year day hr min sec Bmag Bgsm_x Bgsm_y Bgsm_z -s 2003 331 -e 2003 340 > foo`

…where the filename foo will contain the ASCII text results of the request; in this

case, 16 second MAG data for days 2003/331 thru 2003/340, returning data consisting of records with the fields: year, day of year, hour, minute, second, <|B|>, Bgsm_x Bgsm_y, and Bgsm_z..

Plotting Options

It is possible to obtain a simple GIF time-series plot of the data, instead of the data itself. Output is an HTML message with the URL of the plot embedded, unless the "url" argument is supplied (see below).

Plotting arguments

`plot` – use this argument to obtain a GIF plot instead of data.

The following arguments are optional and may be used to modify the plot appearance:

`overplot` or `stacked` (default)

`logarithmic` or `linear` (default)

`multirange` or `samerange` (default)

`x_year`, `x_epoch`, `x_records` or `x_day` (default)

`-title` "*plot-title*"

`url` – Output is a URL to the plot, with no enveloping HTML message.

Note: for a plot request, the time variables (year, day, hr, min, sec, etc.) should not generally be included on the command line, since they will be plotted along with the data.

Plotting example:

afs.pl plot x_day url swics 1hr FetoO -s 03 330 -e 03 332

The libwww-perl Library

The libwww-perl collection is a set of Perl modules which provides a simple and consistent application programming interface to the World-Wide Web. The main focus of the library is to provide classes and functions that allow you to write WWW clients. The library also contain modules that are of more general use and even classes that help you implement simple HTTP servers.

The latest libwww-perl package as well as perl itself are available on CPAN (Comprehensive Perl Archive Network). This is also the place to obtain the other packages that libwww-perl depends upon (URI, HTML::Parser, MIME::Base64, Digest::MD5, and Net::FTP from libnet).