GeoWS Well Structured Text Format

Version: 2015-3-18

Purpose: Specify a common system of annotations and rules for data in structured text formats. An important factor is readability for both humans and machines. Simplicity is considered key for adoption and use. Data compliant with this specification is primarily targeted at delivery as data streams from a GeoWS web services. Ideally, existing structured text data would need very minimal modification, perhaps just additional headers, to be compliant.

Requirements and Assumptions:

1. **Readability** by both humans and computers is very important.
2. **UTF-8** is the text encoding.
3. A "**line**" in a dataset is a string of text ending with one or more newline or carriage return characters.
    a. Lines starting with # are treated as comments, unless a **keyword** is present.
    b. Lines starting with # can occur anywhere in the data stream.
    c. Lines without leading # are treated as delimited data.
4. A header line with a **keyword** takes the following form:

    # keyword : value
    or as a list of values:
    # keyword : value1, value2, value3

    A line beginning with a '#', followed by zero or more whitespace characters, followed by the keyword itself, followed by zero or more whitespace characters, followed by a ':', followed by the keyword value.
    A POSIX extended regular expression to identify a keyword and value:

    '^#\s*(keyword)\s*:(value)[\r\n]+'
5. **Padding keyword values** with spaces. Keyword values that are singular or in a list may be padded with white space that is not considered part of the value, all whitespace before and after a value should be trimmed by a reader. These pairs of headers are equivalent:

    # keyword : value \n
    #keyword:value\n

    or

    # keyword : value1 , value2 of apples , value3
    # keyword:value1,value2 of apples,value3
6. **Known keywords** are:
    a. **fields**: names for each column of data (required, no empty values allowed)
    b. **field_unit**: units for each column of data
    c. **field_type**: types for each column, one of 'string','integer','float','datetime'

    d. **field_long_name**: long descriptive field names, ala CF
    e. **field_standard_name**: long descriptive field names from a vocabulary, ala CF
    f. **field_missing**: values used to denote missing values in the data
    g. **delimiter**: single character delimiter for data values
    h. **attribution**: identify attribution information, probably a URL
    i. **standard_name_cv**: identify controlled vocabulary for **field_standard_name**
    j. from CF: **title**, **history**, **institution**, **source**, **comment**, **references**
7. The only required header keyword is **fields** and it should be the first header and logically starts a data set. When multiple datasets occur in a stream the #fields header denotes the start of each set.
8. There are no global fields. Each dataset must be self contained, meaning that it must have all appropriate headers.
9. Values for field names and attributes (all field_* keywords) should use CF attribute names and definitions whenever appropriate and possible.
10. Keyword values for all field attributes (all field_* keywords) are optional and should be left empty if unknown.
11. The delimiter used for lists of keyword values is a comma.
12. The default delimiter for data is a comma.
13. Fields of type ‘datetime’ must be in an ISO 8601 format. The form of ‘YYYY-MM-DDThh:mm:ss.sss’ is strongly recommended, with the time portion optional for date-only specification and optional time zone designation per ISO 8601.
14. As a minor extension to CF field names for latitude and longitude, the producer and consumer should recognize any field names that begin with "lat" or "lon" (case-insensitive) respectively as latitude and longitude. In addition " lat" or " lon" anywhere in the field name, e.g. Geodetic Longitude, will also be recognized. Example field names: lon, long, longitude, Longitude, LON, LONG, LONGITUDE, Geodetic Longitude, lonnad27, lonnad83 are all accepted as longitude. Latitude has the same rules respectively.
15. Delimiters within delimiters, aka using commas within header value lists
    a. If a comma is needed within a comma-delimited header value, use the hex version of \x2C for comma.

For consideration

Profiles identified by the keyword **profile** could be defined to identify common sets of fields in a given data set, some required and some optional. This would provide a reader with a known set of parameters in a result set. Prototype profile definitions are included after the examples.

Examples:

Example 1: UNAVCO Example of **fields** and **fields_index**

 # fields: ID, station_name, latitude, longitude, ellip_height, session_start_time, session_stop_time
 # field_unit: UTF-8, UTF-8, degrees_north, degrees_east, meters, UTC, UTC
 # field_type: string, string, float, float, float, datetime, datetime
 # attribution: http://www.unavco.org/community/policies_forms/attribution/attribution.html
 # GeodeticDatum: ITRF2008 epsg:1061
 # Ellipsoid: GRS 1980 epsg:7019
 # Ellipsoidal Coordinate System: EllipsoidalCS epsg:6423
 # Axes: Geodetic longitude, Geodetic latitude, Ellipsoidal height. Orientations: east, north, up.
 # Units of Measure: decimal degrees, decimal degrees, meters
ASBU,Astronaut Butte,43.8206,-121.3685,1234,2011-08-18T00:00:00,2015-02-16T23:59:45
CIHL,Cinder Hill,43.7509,-121.1487,4567,2011-09-13T16:04:30,2015-02-16T23:59:45
CPCO,Central Pumice
Cone,43.7221,-121.2332,4321,2011-08-18T00:00:00,2012-03-05T12:02:30
CPCO,Central Pumice
Cone,43.7221,-121.2332,222,2012-06-14T00:00:00,2012-09-25T23:59:45
CPCO,Central Pumice
Cone,43.7221,-121.2332,999,2012-09-26T19:28:45,2013-06-10T22:11:15

Example 2: IRIS

Station metadata example
 # fields: Network, Station, Latitude, Longitude, Elevation, SiteName, StartTime, EndTime
 # field_unit: ASCII, ASCII, degrees_north, degrees_east, meters, UTC, UTC
 # field_type: string, string, float, float, float, string, datetime, datetime
 # delimiter: |
IU|ANMO|34.9459|-106.4572|1850.0|Albuquerque, New Mexico,
USA|1989-08-29T00:00:00|1995-07-14T00:00:00
IU|ANMO|34.9459|-106.4572|1850.0|Albuquerque, New Mexico,
USA|1995-07-14T00:00:00|2000-10-19T16:00:00

Minimal IRIS Station example:
 # fields: Network, Station, Latitude, Longitude, Elevation, SiteName, StartTime, EndTime
 # delimiter: |
IU|ANMO|34.9459|-106.4572|1850.0|Albuquerque, New Mexico,
USA|1989-08-29T00:00:00|1995-07-14T00:00:00
IU|ANMO|34.9459|-106.4572|1850.0|Albuquerque, New Mexico,
USA|1995-07-14T00:00:00|2000-10-19T16:00:00

Event (earthquake) parameter example:

 # fields: EventID, Time, Latitude, Longitude, Depth/km, Author, Catalog, Contributor, ContributorID, MagType, Magnitude, MagAuthor, EventLocationName
 # delimiter: |
3954686|2010-03-01T06:27:32|38.251|69.4919|12.0|ISC|ISC|ISC|00301439|mb|4.3|NNC|TAJIKISTAN
3954685|2010-03-01T06:25:56|37.26|138.91|9.0|JMA|ISC|ISC|15237974|mb|0.5|JMA|NEAR WEST COAST OF HONSHU, JAPAN

Example #: UNIDATA example

Example #: LDEO example

Example #: CUAHSI example

The are presently based on an earlier version.

Observation Catalog:

```
#fields=siteCode[type='string'],siteName[type='string'],latitude[unit='degrees'],longitude[unit='degrees'],variableCode[type='string'],startTime[type='date' format='yyyy-MM-dd],endTime[type='date' format='yyyy-MM-dd]
#profile: observation catalog
LittleBearRiver:USU-LBR-Mendon,"Little Bear River at Mendon Road near Mendon, Utah",41.718473,-111.946402,LittleBearRiver:USU3,2005-08-04,2015-02-26
```

Data Values.

```
#fields=siteCode[type='string'],variableCode[type='string'],dateTime[type='date' format='yyyy-MM-dd HH:mm:ss'],dataValue
#profile: timeseries
LittleBearRiver:USU-LBR-Mendon,LittleBearRiver:USU10,2010-01-01 00:00:00,1.9
LittleBearRiver:USU-LBR-Mendon,LittleBearRiver:USU10,2010-01-01 00:30:00,1.8
LittleBearRiver:USU-LBR-Mendon,LittleBearRiver:USU10,2010-01-01 01:00:00,1.9
```

Prototype profile definitions

Profiles-- profiles are a suggested set of minimum field listings.

b. station - a set of point locations
   i. stationCode (R, ‘string’)
   ii. latitude (R, decimal degrees)
   iii. longitude (R, decimal degrees)
   iv. stationName (O, ‘string’)
   v. stationType (o, ‘string’)

c. observation catalog - a listing of observations for stations with time ranges of availability.
   i. stationCode (R, ‘string’)
   ii. latitude (R, decimal degrees)
   iii. longitude (R, decimal degrees)
   iv. variableCode (R, ‘string’)
   v. startTime (R,`'date'`)
   vi. endTime (R, `'date'`)
   vii. stationName (O)
   viii. variableName (O)

d. timeseries
   i. stationCode (R, ‘string’)
   ii. variableCode (R, ‘string’)
   iii. dateTime(R, `'datetime'`)
   iv. dataValue (R)
   v. stationName (O, ‘string’)
   vi. variableName (O, ‘string’)
   vii. latitude (O, decimal degrees)
   viii. longitude (O, decimal degrees)

e. fielded timeseries
   i. stationCode (R, ‘string’)
   ii. dateTime(R, `'datetime'`)
   iii. field name of property 1
   iv. field name of property n
   v. stationName (O, ‘string’)
   vi. latitude (O, decimal degrees)
   vii. longitude (O, decimal degrees)