A Standard for Exchangeable Magnetotelluric Data and Metadata

Working Group for Data Handling and Software - PASSCAL Magnetotelluric Program[1]

[1]Portable Array Seismic Studies of the Continental Lithosphere, Incorporated Research Institutions for Seismology

Version 0.0.1b – May 2020*

Contents

* ***Corresponding Authors:***
Jared Peacock (jpeacock@usgs.gov)
Andy Frassetto (andy.frassetto@iris.edu)

1 Introduction

Researchers using magnetotelluric (MT) methods lack a standardized format for storing time series data and metadata. Commercially available MT instruments produce data in formats that range from proprietary binary to ASCII, and recent datasets from the U.S. MT community have utilized institutional formats or heavily adapted formats like miniSEED. In many cases, the available metadata for these time series are incomplete and only loosely standardized, and overall these datasets are not "user friendly". This lack of resources impedes the exchange and broader use of these data beyond a small community of specialists.

The IRIS PASSCAL MT facility maintains a pool of MT instruments that are freely available to U.S. Principal Investigators (PIs). Datasets collected with these instruments are subject to data sharing requirements, and an IRIS working group advises the development of sustainable data formats and workflows for this facility. Following in the spirit of the standard created for MT transfer function datasets, this document outlines a new metadata standard for MT time series. This standard is a key pillar of MTH5, a new data format which we propose for the international community of MT practitioners. Further information regarding MTH5 will be available later in 2020.

The Python 3 module written for these standards are found at `https://github.com/kujaku11/MTarchive/tree/tables`.

2 General Structure

The metadata for a full MT dataset are structured to cover details from single channel time series to the full survey. For simplicity each of the different scales of an MT survey and measurements have been categorized starting from largest to smallest (Figure 1. These categories are: `Survey`, `Station`, `Run`, `DataLogger`, `Electric Channel`, `Magnetic Channel`, and `Auxiliary Channels`. Each of these are described in subsequent sections.

2.1 Metadata Keyword Format

The metadata key names should be self explanatory and they are structured as follows: `{category}.{name}`, where:

- `category` refers to a metadata category that has common parameters, such as `location` which will have a latitude, longitude, and elevation $\longrightarrow$ `location.latitude`, `location.longitude`, and `location.elevation`. These can be nested, for example `positive.location.latitude`
- `name` is the description name, where words should be separated by an underscore, e.g. `data_quality`.

Alternatively, the metadata names can be nested under category headings as commonly done in XML or JSON formats. See examples below for various flavors of ways to represent the metadata.

Table 1: Permissible values for data types

Data Type
String
Double (float)
Integer
Boolean

2.2 Formatting Standards

Specific and required formatting standards for location, time and date, and angles are defined below.

2.2.1 Time and Date Format

All time and dates are given as an ISO formatted date-time string in the UTC time zone. The ISO date time format is `YYYY-MM-DDThh:mm:ss.ms+00:00`. Milliseconds can be accurate to 6 decimal places. Dates are formatted `YYYY-MM-DD`.

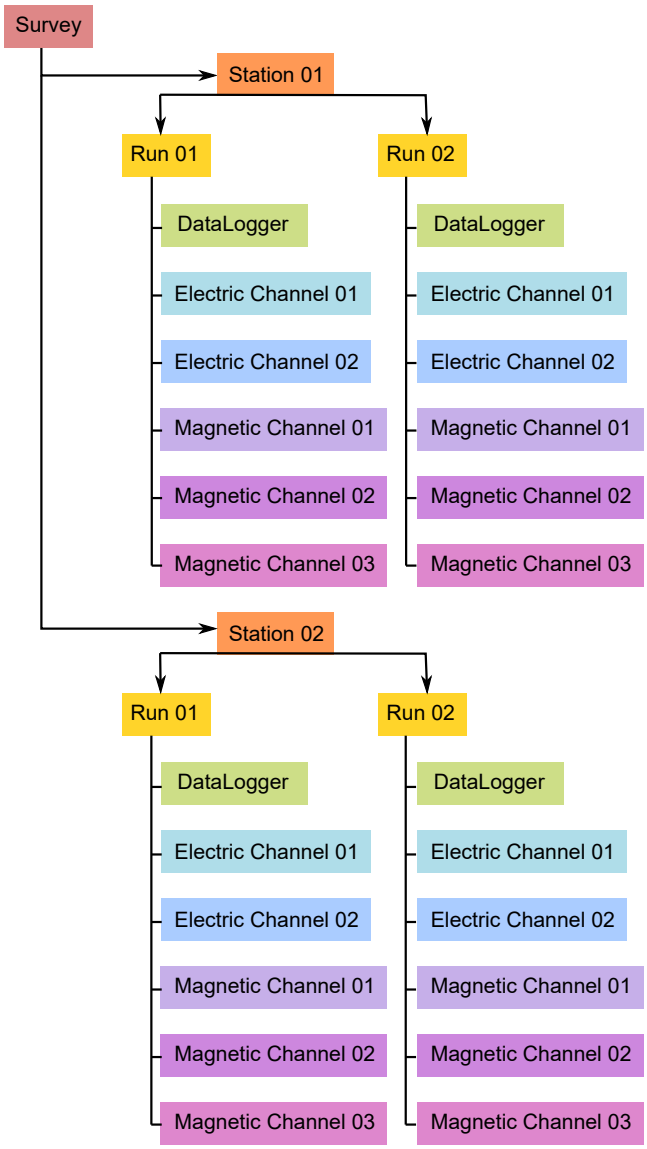

Figure 1: Schematic of a MT time series file structure with appropriate metadata.

2.2.2 Location

All latitude and longitude locations are given in decimal degrees in a well known datum.

- All latitude values must be $< |90|$ and all longitude values must be $< |180|$.
- Elevation and other distance values are given in meters.
- Datum should be one of the well known datums, WGS84 is preferred, but others are acceptable.

2.2.3 Angles

All angles of orientation are given in degrees. Orientation of dipoles and magnetometers are relative to `station_orientation`. Otherwise angles are assumed to be clockwise positive from Geographic North = 0.

2.3 Units

Units should all be from the metric system. Abbreviations and full names are acceptable, for example `mV` and `millivolts`. Below are a summory of common acceptable units:

Table 2: Acceptable units

Measurement Type	**Unit Long Name**	**Unit Short Name**
Angles	degrees	deg
Distance	meters	m
Latitude/Longitude	decimal degrees	deg
Resistance	Ohms	Ohms
Resistivity	Ohm-meters	Ohm-m, Ohmm
Temperature	Celsius	C
Time	seconds	s
Voltage	Volts	V

3 Survey

A survey describes an entire dataset that covers a specific time span and region. This may include multiple PIs in multiple data collection episodes but should be confined to a specific experiment. The `Survey` metadata category describes the general parameters of the survey.

Table 3: Attributes for Survey category

Metadata Key	**Description**	**Type**	**Required**
project	alphanumeric name for the project e.g USGS-GEOMAG	string	compulsory
survey	long name of the survey, e.g. Magnetotelluric investigations of Mammoth volcanic system, California	string	optional
net_code	network code given by PASSCAL/IRIS/FDSN	string	compulsory
start_date	start date of survey [UTC]	string	compulsory
end_date	end date of survey [UTC]	string	compulsory
northwest_corner.latitude	location of northwest corner of survey [degrees]	float	compulsory
northwest_corner.longitude	location of northwest corner of survey [degrees]	float	compulsory
southeast_corner.latitude	location of southeast corner of survey [degrees]	float	compulsory
southeast_corner.longitude	location of southeast corner of survey [degrees]	float	compulsory
datum	datum of latitude and longitude coordinates [WGS84]	string	compulsory
geographic_location	geographic location of survey in general terms	string	optional
country	country/countries survey located in	string	optional
summary	summary paragraph of survey	string	compulsory
notes	notes about survey	string	optional
acquired_by.author	principal investigator(s) responsible for survey	string	compulsory
acquired_by.organization	organization(s) associated with survey	string	compulsory
acquired_by.email	email address of PI(s)	string	compulsory
acquired_by.url	url(s) of organization(s)	string	compulsory
release_status	release status [open | on request | propriatary |...]	string	compulsory
conditions_of_use	condition of use information information including licensing	string	optional
citation_data_set.doi	citation dataset doi number	string	compulsory
citation_journal.doi	citation journal doi	string	optional

3.1 Example Survey JSON Input String


```
{"survey": {
    "name": "Long Valley, CA",
    "id": "Casa Diablo",
    "net_code": "XE",
    "start_date": "2020-01-01",
    "end_date": "2021-01-01",
    "northwest_corner": {
        "latitude": 37.5,
        "longitude": 122.0
    },
    "southeast_corner": {
        "latitude": 36.5,
        "longitude": -121.15
    },
    "datum": "WGS84",
    "location": "Mammoth, CA",
    "country": "USA",
    "summary": "This survey is meant to image the magmatic and hydrothermal systems.",
    "notes": "Had complications due to snow",
    "release_status": "open",
    "conditions_of_use": "condition of use information including licensing",
    "citation_dataset": {
        "doi": "citation dataset doi number"
    },
    "citation_journal": {
        "doi": "citation journal doi"
    },
    "acquired_by": {
        "author": "M. Tee, T. Luric, S. Spot, and A. Borealis",
        "organization": "MT Gurus",
        "url": "mt_guru.com",
        "email": "mtee@guru.com"
    }
    }
}
```

4 Station

A station encompasses a single site where data are collected. If the location changes during a run, then a new station should be created. If the sensors, cables, data logger, battery are replaced during a run but the station remains stations, then this can be recorded in the `Run` metadata but does not require a new station entry.

Table 4: Attributes for Station category

Metadata Key	**Description**	**Type**	**Required**
sta_code	5 char name A-Z; 1-9 for station	string	compulsory
geographic_name	geographic name station site	string	compulsory
location.latitude	longitude location [degrees (hh.mmss)]	float	compulsory
location.longitude	latitude location [degrees (hh.mmss)]	float	compulsory
location.elevation	elevation [m]	float	compulsory
location.declination.value	declination value	float	compulsory
location.declination.epoch	declination epoch	string	compulsory
location.declination.model	declination model	string	compulsory
notes	any notes about station	string	optional
datum	datum for lat, lon location	string	compulsory
start	start time and date of data logging [ISO UTC]	string	compulsory
end	stop time and date of data logging [ISO UTC]	string	compulsory
num_channels	number of channels recording	int	compulsory
channels_recorded	"list of channels recorded [EX, EY, HX, HY, HZ | ...]"	string	compulsory
data_type	type of data collected [BB | LP | AMT | Combo | ...]	string	compulsory
station_orientation	orientation coordinate system [geographic | channel-measurement specific | ...]	string	compulsory
orientation_method	[compass | differential GPS | gyroscope |...]	string	optional
acquired_by.author	person(s) operating station	string	compulsory
acquired_by.email	email of lead station operator	string	compulsory
provenance.creation_time	creation time of time series data for storing	string	compulsory
provenance.software.name	name of software used to store time series	string	compulsory
provenance.software.version	version of software used to store time series	string	compulsory
provenance.submitter.author	name of person or group submitting archive data	string	compulsory
provenance.submitter.organization	name of organization or institution submitting archive data	string	compulsory
provenance.submitter.url	url of group submitting archive data	string	compulsory
provenance.submitter.email	email of person or group submitting archive data	string	compulsory
provenance.notes	any notes on the history of the data	string	optional
provenance.log	log of any changes made to time series data	string	optional

4.1 Example Station JSON Input String


```
{
 "sta_code": "MNP01",
 "name": "Mojave National Preserve Hole-in-the-rock",
 "location.latitude": 35.0,
 "location.longitude": -117.0,
 "location.elevation": 1200,
 "location.declination.value": "11.5",
 "location.declination.epoch": "declination epoch",
 "location.declination.model": "WMM2019-2024",
 "notes": "Donkeys chewed both electric channels",
 "datum": "WGS84",
 "start": "2020-01-01T12:00:00.0000+00:00",
 "end": "2020-01-12T12:00:00.0000+00:00",
 "num_channels": 5,
 "channels_recorded": "[EX, EY, HX, HY, HZ]",
 "data_type": "BB & LP",
 "station_orientation": "geographic",
 "orientation_method": "compass",
 "acquired_by.author": "M. Tee and A. Borealis",
 "acquired_by.email": "m.tee@guru.com",
 "provenance.creation_time": "2020-05-01T12:00:00.0000+00:00",
 "provenance.software.name": "MTH5",
 "provenance.software.version": "1.0.0",
 "provenance.software.author": "IRIS",
 "provenance.submitter.author": "M. Tee",
 "provenance.submitter.organization": "MT Gurus",
 "provenance.submitter.url": "mt_guru.com",
 "provenance.submitter.email": "m.tee@guru.com",
 "provenance.notes": "Electrics are good until 2020-01-10",
 "provenance.log": "2020-05-02T12:00:00+00:00: The data rotated with updated
                    declination of N13.78W."
}
```

5 Run

A run represents data collected at a single station with a single sampling rate. If the dipole length or other such station parameters are changed between runs, this would require adding a new run. If the station is relocated then a new station should be created.

Table 5: Attributes for Run category

Metadata Key	Description	Type	Required
id	run ID	string	compulsory
notes	notes on run	string	optional
start	start date and time of data logging [ISO UTC]	string	compulsory
end	stop date and time of data logging [ISO UTC]	string	compulsory
sampling_rate	sampling rate of run (samples.second)	float	compulsory
num_channels	number of channels recorded	int	compulsory
channels_recorded	list of channels recorded [[EX, EY, HX, HY] | ...]	string	compulsory
data_type	type of data collected [BB | LP | AMT | Combo | ...]	string	compulsory
acquired_by.author	person(s) responsible for run	string	compulsory
acquired_by.email	email of lead run operator	string	compulsory
provenance.notes	any notes on the history of the data	string	optional
provenance.log	log of any changes made to time series data	string	optional

5.1 Example Run XML Input String

```
<run>
    <id>MNP02_b</id>
    <notes>Changed north electrode</notes>
    <start>2020-01-02T15:30:00+00:00</start>
    <end>2020-01-05T07:05:30+00:00</end>
    <sampling_rate type="float" units="samples per second">256.0</sampling_rate>
    <channels_recorded>[EX, EY, HX, HY, HZ]</channels_recorded>
    <data_type>BB</data_type>
    <acquired_by>
        <author>T. Luric</author>
        <email>t.lurric@guru.com</email>
    </acquired_by>
    <provenance>
         <notes>Near a powerline and HZ is clipped</notes>
         <log>2020-05-01T12:00:00+00:00: Clipped data in HZ replaced with zeros by T. Luric.</log>
    </provenance>
</run>
```

6 Data Logger

Data logger is a the digital acquisition system used to collect time series data at a single station for a single run. `DataLogger` metadata includes the type of data logger, timing system, firmware, number of channels, calibrations, and power source.

Table 6: Attributes for DataLogger category

Metadata Key	**Description**	**Type**	**Required**
manufacturer	manufacturer name	string	compulsory
model	model name	string	compulsory
serial	serial number	string	compulsory
notes	notes about data logger	string	compulsory
timing_system.type	type of timing system [GPS | internal | ...]	string	compulsory
timing_system.drift	any drift in internal clock [seconds]	float	compulsory
timing_system.uncertainty	uncertainty associated with internal clock [seconds]	float	compulsory
timing_system.notes	notes on timing system	string	optional
firmware.version	firmware version	string	compulsory
firmware.date	date on firmware	string	compulsory
firmware.author	author of firmware	string	optional
n_channels	number of channels	int	compulsory
n_channels_used	number of channels used	int	compulsory
power_source.type	power source type [Pb-acid battery | solar panel | Li battery | ...]	string	compulsory
power_source.id	power source id	string	optional
power_source.voltage.start	starting voltage of power source	float	compulsory
power_source.volage.end	ending voltage of power source	float	compulsory
power_source.notes	notes on power source	string	optional

6.1 Example DataLogger JSON Input String

```
{
 "manufacturer": "MT ‘r Us",
 "model": "Broadband 2000",
 "serial": "0128947850230",
 "notes": "Intern dropped the data logger on a shovel.",
 "timingystem.type": "GPS",
 "timingystem.drift": 0,
 "timingystem.uncertainty": 0.0016,
 "timingystem.notes": "only works when sky is clear",
 "firmware.version": "1.0",
 "firmware.date": "2020-01-01",
 "firmware.author": "R. Phase",
 "n_channels": 5,
 "n_channels_used": 4,
 "powersource.id": "battery 10"
 "powersource.type": "solar panel and battery",
 "powersource.voltage.start": 13.1,
 "powersource.voltage.end": 12.0,
 "powersource.notes": "Overcast all day reduced recharging"
}
```

7 Electric Channel

Electric channel refers to a dipole measurement of the electric field for a single station for a single run.

Table 7: Attributes for Electric category

Metadata Key	**Description**	**Type**	**Required**
dipole_length	length of dipole [m]	float	compulsory
channel_number	channel number [1 | 2 | 3 | 4 | 5 | 6 |...]	int	compulsory
component	[Ex | Ey | Ez]	string	compulsory
azimuth	azimuth of dipole N = 0, E = 90 [degrees]	float	compulsory
positive.id	sensor id number	string	compulsory
positive.latitude	positive sensor location latitude [degrees (hh.mmss)]	float	optional
positive.longitude	positive sensor location longitude [degrees (hh.mmss)]	float	optional
positive.elevation	positive sensor location elevation [m]	float	optional
positive.datum	positive datum for location [WGS84]	string	optional
positive.type	type of electric sensor [Ag-AgCl | Pb-PbCl | ...]	string	compulsory
positive.manufacturer	electric sensor manufacturer	string	compulsory
positive.notes	notes on electric sensor	string	optional
negative.id	sensor id number	string	compulsory
negative.longitude	negative sensor location latitude [degrees (hh.mmss)]	float	optional
negative.latitude	negative sensor location longitude [degrees (hh.mmss)]	float	optional
negative.elevation	negative sensor location elevation [m]	float	optional
negative.datum	negative datum for location [WGS84]	string	optional
negative.type	type of electric sensor [Ag-AgCl | Pb-PbCl | ...]	string	compulsory
negative.manufacturer	electric sensor manufacturer	string	compulsory
negative.notes	notes on electric sensor	string	optional
contact_resistance_1.start	contact resistance at beginning of measurement, positive polarity [Ohm]"	float	optional
contact_resistance_2.start	contact resistance at beginning of measurement, negative polarity [Ohm]	float	optional
contact_resistance_1.end	contact resistance at end of measurement, positive polarity [Ohm]	float	optional
contact_resistance_2.end	contact resistance at end of measurement, negative polarity [Ohm]	float	optional
ac.start	AC at start of measurement [V]	float	optional
ac.end	AC at end of measurement [V]	float	optional
dc.start	DC at start of measurement [V]	float	optional
dc.end	DC at end of measurement [V]	float	optional

Table 8: Attributes for Electric category continued

Metadata Key	**Description**	**Type**	**Required**
units	units of electric field data [counts | mV/km | ...]	string	compulsory
sample_rate	sample rate of electric channel (samples.second)	float	compulsory
notes	notes about electric field measurement	string	optional
data_quality.rating	data quality rating based on some sort of statistic	integer	optional
data_quality.warning_notes	any warnings about data quality	string	optional
data_quality.warning_flags	a value flagging bad data	float	integer
data_quality.author	person who did QA/QC on data	string	optional
filter.name	filter name in filter table, can be a list. Needs to be ordered in which filters were applied	string	optional
filter.notes	any notes on the filtering	string	optional
filter.applied_b	have filters been applied [True | False]	string	compulsory

7.1 Example Electric Channel JSON Input String


```
{
 "dipole_length": 59.7,
 "channel_number": "1",
 "component": "EX",
 "azimuth": 0,
 "positive.id": "101",
 "positive.latitude": 35.5578,
 "positive.longitude": -117.38754,
 "positive.elevation": 103.4,
 "positive.datum": "WGS84",
 "positive.type": "Ag-AgCl",
 "positive.manufacturer": "Zaps",
 "positive.notes": "Sitting on the shelf since last year",
 "negative.id": "102",
 "negative.latitude": 35.5588,
 "negative.longitude": -117.38754,
 "negative.elevation": 105.8,
 "negative.datum": "WGS84",
 "negative.type": "Ag-AgCl"
 "negative.manufacturer": "Zaps",
 "negative.notes": "Sitting on the shelf since last year",
 "contact_resistance_1.start": 1200.0,
 "contact_resistance_2.start": 1210.0,
 "contact_resistance_1.end": 1205.0,
 "contact_resistance_2.end": 1205.0,
 "ac.start": 0.03,
 "ac.end": 0.04,
 "dc.start": 0.001,
 "dc.end": 0.002,
 "units": "counts",
 "sample_rate": 256,
 "notes": "cables chewed on 2020-01-07",
 "data_quality.rating": 3,
 "data_quality.warning_notes": "cables chewed 2020-01-07",
 "data_quality.warning_flags": "Nan",
 "data_quality.author": "Q. Sea",
 "filter.name": "[counts2mv, datalogger024]",
 "filter.notes": "notes on filters applied",
 "filter.applied_b": "true"
}
```

8 Magnetic Channel

A magnetic channel is a recording of one component of the magnetic field at a single station for a single run.

Table 9: Attributes for Magnetic category

Metadata Key	**Description**	**Type**	**Required**
sensor.type	type of magnetic sensor [Induction Coil | flux gate | ...]	string	compulsory
sensor.manufacturer	magnetic sensor manufacturer	string	compulsory
sensor.notes	notes on sensor	string	compulsory
sensor.id	sensor id number	string	compulsory
channel_number	channel number [1 | 2 | 3 | 4 | 5 | 6 |...]	int	compulsory
component	[Hx | Hy | Hz]	string	compulsory
azimuth	azimuth in `station\_coordinates` [degrees]	float	compulsory
longitude	sensor longitude degrees	float	compulsory
latitude	sensor latitude in degrees	float	compulsory
elevation	sensor elevation in meters	float	compulsory
datum	datum for location [WGS84 | ...]	string	compulsory
units	units of magnetic field data [counts | mV | ...]	string	compulsory
sample_rate	sample rate of magnetic channel (samples.second)	float	compulsory
h_field_min.start	minimum h-field value at beginning of measurement	float	optional
h_field_max.start	maximum h-field value at beginning of measurement	float	optional
h_field_min.end	minimum h-field value at end of measurement	float	optional
h_field_max.end	maximum h-field value at end of measurement	float	optional
h_field.units	units of h-field measurement [nT | ...]	string	optional
notes	notes on magnetic field measurments	string	optional
data_quality.rating	data quality rating based on some sort of statistic	integer	optional
data_quality.warning_notes	any warnings about data quality	string	optional
data_quality.warning_flags	a value flagging bad data	integer	optional
data_quality.author	person who did QC.QA on data	string	optional
filter.name	filter name in filter table, can be a list. Needs to be ordered in which filters were applied	string	optional
filter.notes	any notes on the filtering	string	optional
filter.applied_b	have filters been applied [True | False]	string	compulsory

8.1 Example Magnetic Channel JSON Input String


```
{
 "sensor.type": "Induction Coil",
 "sensor.manufacturer": "MT ‘r Us",
 "sensor.notes": "new coil",
 "sensor.id": "2149",
 "channel_number": 5,
 "component": "HZ",
 "azimuth": 90,
 "longitude": -117.0,
 "latitude": 45.0,
 "elevation": 107.4,
 "datum": "WGS84",
 "units": "counts",
 "sample_rate": 256,
 "h_field_min.start": -10,
 "h_field_max.start": 10,
 "h_field_min.end": -9,
 "h_field_max.end": 9,
 "h_field.units": "nT",
 "notes": "not buried all the way ",
 "data_quality.rating": 4,
 "data_quality.warning_notes": "windy during the day",
 "data_quality.warning_flags": 0,
 "data_quality.author": "Q. Sea",
 "filter.name": "[counts2mv, datalogger024, coil2149]",
 "filter.notes": "Calibrated 2018-01-01",
 "filter.applied_b": "[true, false, false]"
}
```

9 Filters

Filters is a table that holds information on any filters that need to be applied to get physical units, and filters that were applied to the data to analyze the signal. This includes calibrations, notch filters, conversion of counts to units, etc. The actual filter will be an array of numbers contained within an array named `name` and formatted according to `type`. The preferred format for a filter is a look-up table which internally can be converted to other formats.

It is important to note that filters will be identified by name and must be consistent throughout the file. Names should be descriptive and self evident. Examples:

- `coil_2284` $\longrightarrow$ induction coil number 2284
- `counts2mv` $\longrightarrow$ conversion from counts to mV
- `e_gain` $\longrightarrow$ electric field gain
- `datalogger_024` $\longrightarrow$ data logger number 24 response
- `notch_60hz` $\longrightarrow$ notch filter for 60 Hz and harmonics
- `lowpass_10hz` $\longrightarrow$ low pass filter below 10 Hz

In each channel there are keys to identify filters that can or have been applied to the data to get an appropriate signal. This can be a list of filter names or a single filter name. An `applied_b` key also exists for the user to input whether that filter has been applied. Can be a single Boolean `true` if all filters have been applied, `false` if none of the filters have been applied. Or can be a list the same length and the filter name list identifying if the filter has been applied. `name: "[counts2mv, notch60hz, e_gain]"` and `applied_b: "[true, false, true]"`.

Table 10: Attributes for Filters

Metadata Key	**Description**	**Type**	**Required**
type	type of filter [look up | poles-zeros | converter | FIR | ...]	string	compulsory
name	unique name for the filter such that it is easy to query	string	compulsory
units_in	units of data going in [counts | mV/km | ...]	string	compulsory
units_out	units of data coming out [counts | mV/km | ...]	string	compulsory
calibration_date	date of calibration	string	compulsory
notes	any notes on the filtering	string	optional

9.1 Example Filter JSON Input String

```
{
 "type": "look up",
 "name": "coil_8897",
 "unitsn": "mV",
 "units_out": "mV",
 "calibrationate": "2015-07-01",
 "notes": "interpolated from poles and zeros"
}
```

10 Auxiliary Channels

Auxiliary channels include state of health channels, temperature, etc.

Table 11: Attributes for Auxiliary category

Metadata Key	**Description**	**Type**	**Required**
type	type of data recorded [temperature | GPS | ...]	string	compulsory
units	units of magnetic field data [counts | mV | ...]	string	compulsory
channel_num	channel number [1 | 2 | 3 | 4 | 5 | 6 |...]	int	compulsory
component	channel number ['None' | ...]	string	compulsory
sample_rate	sample rate (samples.second)	float	compulsory
notes	any notes on the auxillary channel	string	optional
data_quality.rating	data quality rating based on some sort of statistic	integer	optional
data_quality.warning_notes	any warnings about data quality	string	optional
data_quality.warning_flags	a value flagging bad data	integer	optional
data_quality.author	person who did QC.QA on data	string	optional
filter.name	filter name in filter table, can be a list. Needs to be ordered in which filters were applied	string	optional
filter.notes	any notes on the filtering	string	optional
filter.applied_b	have filters been applied [True | False]	string	compulsory

10.1 Example Auxiliary JSON Input String

```
{
"type": "temperature",
"units": "celsius",
"channel_number": "6",
"component": "None",
"sample_rate": "256",
"notes": "internal data logger temperature"
"data_quality.rating": 4,
"data_quality.warning_notes": "windy during the day",
"data_quality.warning_flags": "Nan",
"data_quality.author": "Q. Sea",
"filter.name": "[counts2c]"
"filter.notes": "Calibrated 2018-01-01",
"filter.applied_b": "true"
}
```