MT Metadata Guide

IRIS-PASSCAL MT Software Development Committee[1]

[1]IRIS

May 13, 2020

Contents

1 Introduction

The magnetotelluric community is relatively small which has led to various formats for storing the time series data. Some type of ASCII format seems to be the most prevalent because before large data sets that was the easiest method of storage. Various binary formats exist, some proprietary and some open like the Scripps format, though efficient, these files lack some critical metadata. Metadata is key to archiving data and as of now there has been no documentation on metadata standards for MT time series data.

IRIS-PASSCAL is adding MT capabilities to their instrument pool and has setup a committee to develop MT metadata standards for archiving time series. What follows are the metadata standards developed by that committee.

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

2 General Structure

The MT metadata standards are structured to cover details from single channel time series to the full MT survey. For simplicity each of the different scales of an MT survey and measurements have been categorized starting from largest scale to smallest scale (Figure 1). These categories are: `Survey`, `Station`, `Run`, `DataLogger`, `Channel` which has different attributes depending on what type of channel it is, magnetic, electric, auxiliary. Each of these will be described in the sections below.

2.1 Metadata Keyword Format

The metadata key names should be self explanatory and they are structured as follows: `{category}.{name}_{type}`, 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_d`, `location.longitude_d`, and `location.elevation_d`.
- `name` is the description name, where words should be separated by an underscore, e.g. `data_quality`.
- `type` is the data type (Table 1). The `string` encompasses basically anything that cannot be represented as a float, integer, or Boolean. See below for further description of certain string types.

Table 1: Permissible values for data types

Data Type	**Label**
String	s
Double (float)	d
Integer	i
Boolean	b

2.2 Formatting Standards

There are certain formatting standards that need to be adhered to, specifically location, time and date, and angles.

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`. Other formats can be input but only ISO format will be output. Internally, all date time strings are converted to a `datetime` object which can output various formats like epoch seconds, which will keep date-times self consistent.

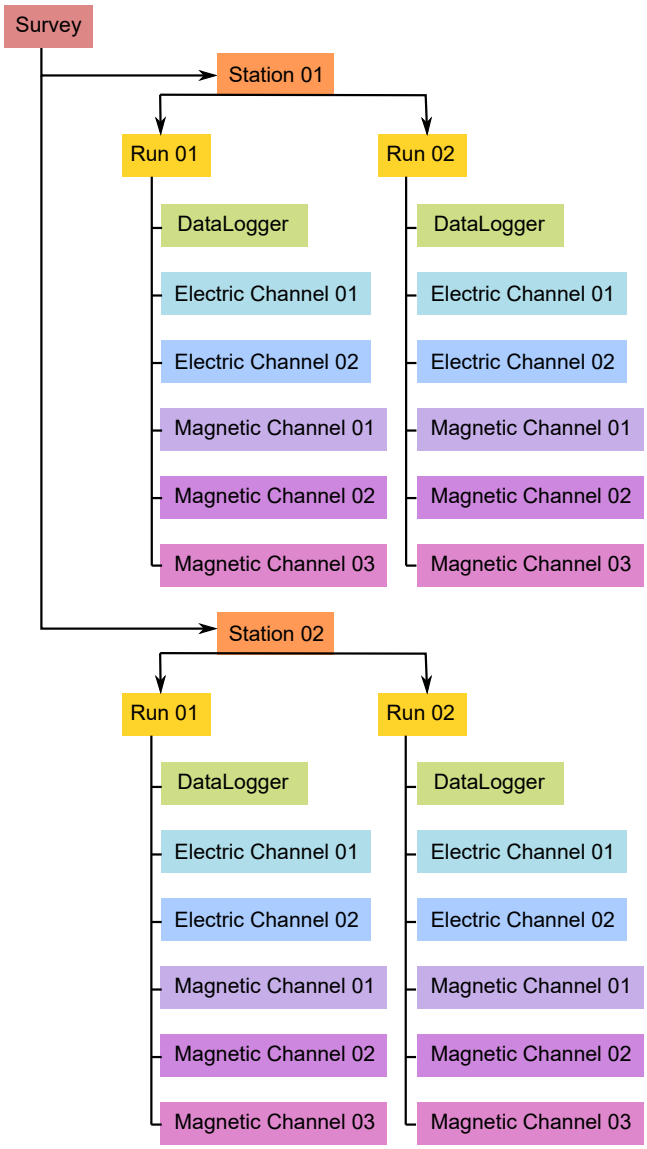

Figure 1: Schematic of a MT time series file structure with appropriate metadata.

2.2.2 Location

All latitude and longitude locations are given in decimal degrees. Other formats can be input but will only be output as decimal degrees.

- 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.

Note: The one exception is sensor location which can be in units relative to the station location?

2.2.3 Angles

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

3 Survey

A survey describes an entire MT survey that covers a specific study area. This may include multiple researchers and research groups or a multi-year campaign, but should be confined to a specific regional area. The `Survey` metadata category describes the general parameters of the survey.

Table 2: Attributes for Survey category

Metadata Key	**Description**	**Type**	**Required**
name_s	name of survey	string	compulsory
id_s	nickname of survey	string	optional
net_code_s	network code given by IRIS	string	compulsory
start_date_s	start date of survey [UTC]	string	compulsory
end_date_s	end date of survey [UTC]	string	compulsory
northwest_corner.latitude_d	location of northwest corner of survey [degrees (hh.mmss)]	float	compulsory
northwest_corner.longitude_d	location of northwest corner of survey [degrees (hh.mmss)]	float	compulsory
southeast_corner.latitude_d	location of southeast corner of survey [degrees (hh.mmss)]	float	compulsory
southeast_corner.longitude_d	location of southeast corner of survey [degrees (hh.mmss)]	float	compulsory
datum_s	"datum of x and y coordinates [WGS84]"	string	compulsory
location_s	location of survey in general terms	string	optional
country_s	country.countries survey located in	string	optional
summary_s	summary paragraph of survey	string	compulsory
notes_s	notes about survey	string	optional
acquired_by.author_s	principal investigator(s) responsible for survey	string	compulsory
acquired_by.organization_s	organization(s) associated with survey	string	compulsory
acquired_by.email_s	email address of PI(s)	string	compulsory
acquired_by.url_s	url(s) of organization(s)	string	compulsory
release_status_s	release status [open | on request | propriatary |...]	string	compulsory
conditions_of_use_s	condition of use information information including licensing	string	optional
citation_dataset.doi_s	citation dataset doi number	string	compulsory
citation_journal.doi_s	citation journal doi	string	optional

3.1 Example Survey JSON String


```
{
 "name_s": "Long Valley, CA",
 "id_s": "Casa Diablo",
 "net_code_s": "network code given by IRIS",
 "start_date_s": "2020-01-01",
 "end_date_s": "2021-01-01",
 "northwest_corner.latitude_d": 37.5,
 "northwest_corner.longitude_d": 122,
 "southeast_corner.latitude_d": 36.5,
 "southeast_corner.longitude_d": -121.15,
 "datum_s": "WGS84",
 "location_s": "Mammoth, CA",
 "country_s": "USA",
 "summary_s": "This survey is meant to image the magmatic and hydrothermal systems.",
 "notes_s": "Had complications due to snow",
 "acquired_by.author_s": "M. Tee, T. Luric, S. Spot, and A. Borealis",
 "acquired_by.organization_s": "MT Gurus",
 "acquired_by.email_s": "mtee@guru.com",
 "acquired_by.url_s": "mt_guru.com",
 "release_status_s": "open",
 "conditions_of_use_s": "condition of use information information including licensing",
 "citation_dataset.doi_s": "citation dataset doi number",
 "citation_journal.doi_s": "citation journal doi",
}
```

4 Station

A station is a single location where MT data are collected, if the location of the station is moved during a run, then a new station should be created. If the sensors, cables, data logger, battery were replaced during a run but the station remained at the same location, that can be recorded in the `Run` metadata, but is still considered a station.

Table 3: Attributes for Station category

Metadata Key	**Description**	**Type**	**Required**
sta_code_s	5 char name of station	string	compulsory
name_s	name station site	string	compulsory
latitude_d	longitude location [degrees (hh.mmss)]	float	compulsory
longitude_d	latitude location [degrees (hh.mmss)]	float	compulsory
elevation_d	elevation [m]	float	compulsory
notes_s	any notes about station	string	optional
datum_s	datum for lat, lon location	string	compulsory
start_s	start time and date of data logging [ISO UTC]	string	compulsory
end_s	stop time and date of data logging [ISO UTC]	string	compulsory
num_channels_i	number of channels recording	int	compulsory
channels_recorded_s	"list of channels recorded [EX, EY, HX, HY, HZ | ...]"	string	compulsory
data_type_s	type of data collected [BB | LP | AMT | Combo | ...]	string	compulsory
declination.value_d	declination value	float	compulsory
declination.units_s	declination units [degrees]	string	compulsory
declination.epoch_s	declination epoch	string	compulsory
declination.model_s	declination model	string	compulsory
station_orientation_s	orientation coordinate system [geographic | channel-measurement specific | ...]	string	compulsory
orientation_method_s	[compass | differential GPS | gyroscope |...]	string	optional
acquired_by.author_s	person(s) operating station	string	compulsory
acquired_by.email_s	email of lead station operator	string	compulsory
provenance.creation_time_s	creation time of time series data for storing	string	compulsory
provenance.software.name_s	name of software used to store time series	string	compulsory
provenance.software.version_s	version of software used to store time series	string	compulsory
provenance.submitter.author_s	name of person or group submitting archive data	string	compulsory
provenance.submitter.organization_s	name of organization or institution submitting archive data	string	compulsory
provenance.submitter.url_s	url of group submitting archive data	string	compulsory
provenance.submitter.email_s	email of person or group submitting archive data	string	compulsory
provenance.notes_s	any notes on the history of the data	string	optional
provenance.log_s	log of any changes made to time series data	string	optional

4.1 Example Station JSON String


```
{
 "sta_code_s": "MNP01",
 "name_s": "Mojave National Preserve Hole-in-the-rock",
 "latitude_d": 35.0,
 "longitude_d": -117.0,
 "elevation_d": 1200,
 "notes_s": "Donkeys chewed both electric channels",
 "datum_s": "WGS84",
 "start_s": "2020-01-01T12:00:00.0000+00:00",
 "end_s": "2020-01-12T12:00:00.0000+00:00",
 "num_channels_i": 5,
 "channels_recorded_s": "[EX, EY, HX, HY, HZ]",
 "data_type_s": "BB & LP",
 "declination.value_d": "11.5",
 "declination.units_s": "degrees",
 "declination.epoch_s": "declination epoch",
 "declination.model_s": "WMM2019-2024",
 "station_orientation_s": "geographic",
 "orientation_method_s": "compass",
 "acquired_by.author_s": "M. Tee and A. Borealis",
 "acquired_by.email_s": "m.tee@guru.com",
 "provenance.creation_time_s": "2020-05-01T12:00:00.0000+00:00",
 "provenance.software.name_s": "MTH5",
 "provenance.software.version_s": "1.0.0",
 "provenance.software.author_s": "IRIS",
 "provenance.submitter.author_s": "M. Tee",
 "provenance.submitter.organization_s": "MT Gurus",
 "provenance.submitter.url_s": "mt_guru.com",
 "provenance.submitter.email_s": "m.tee@guru.com",
 "provenance.notes_s": "Electrics are good until 2020-01-10",
 "provenance.log_s": "The data was rotated using an updated declination 2020-05-02."
}
```

5 Run

A run is data collected at a single station at a single sampling rate. If the dipole length or other such station parameters are changed between runs that is ok, just make a new run. If the station is relocated then a new station should be created.

Table 4: Attributes for Run category

Metadata Key	**Description**	**Type**	**Required**
id_s	run ID	string	compulsory
notes_s	notes on run	string	optional
start_s	start date and time of data logging [UTC]	string	compulsory
end_s	stop date and time of data logging [UTC]	string	compulsory
sampling_rate_d	sampling rate of run (samples.second)	float	compulsory
num_channels_i	number of channels recorded	int	compulsory
channels_recorded_s	"list of channels recorded [[EX, EY, HX, HY] | ...]"	string	compulsory
data_type _s	type of data collected [BB | LP | AMT | Combo | ...]	string	compulsory
acquired_by.author_s	person(s) responsible for run	string	compulsory
acquired_by.email_s	email of lead run operator	string	compulsory
provenance.notes_s	any notes on the history of the data	string	optional
provenance.log_s	log of any changes made to time series data	string	optional

5.1 Example Run JSON String

```
{
 "id_s": "MNP02_b",
 "notes_s": "Changed north electrode",
 "start_s": "2020-01-02T15:30:00.0000+00:00",
 "end_s": "2020-01-05T07:05:30.0000+00:00",
 "sampling_rate_d": 256,
 "num_channels_i": 5,
 "channels_recorded_s": "[EX, EY, HX, HY, HZ]",
 "data_type_s": "BB",
 "acquired_by.author_s": "T. Luric",
 "acquired_by.email_s": "t.lurric@guru.com",
 "provenance.notes_s": "Near a powerline and HZ is clipped",
 "provenance.log_s": "Clipped data in HZ replaced with zeros 2020-05-01 by T. Luric"
}
```

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 5: Attributes for DataLogger category

Metadata Key	**Description**	**Type**	**Required**
manufacturer_s	manufacturer name	string	compulsory
model_s	model name	string	compulsory
serial_s	serial number	string	compulsory
notes_s	notes about data logger	string	compulsory
timing_system.type_s	type of timing system [GPS | internal | ...]	string	compulsory
timing_system.drift_d	any drift in internal clock	float	compulsory
timing_system.uncertainty_d	uncertainty associated with internal clock	float	compulsory
timing_system.notes_s	notes on timing system	string	optional
firmware.version_s	firmware version	string	compulsory
firmware.date_s	date on firmware	string	compulsory
firmware.author_s	author of firmware	string	optional
n_channels_i	number of channels	int	compulsory
n_channels_used_s	number of channels used	int	compulsory
power_source.type_s	power source type [Pb-acid battery | solar panel | Li battery | ...]	string	compulsory
power_source.id_s	power source id	string	optional
power_source.voltage.start_d	starting voltage of power source	float	compulsory
power_source.volage.end_d	ending voltage of power source	float	compulsory
power_source.notes_s	notes on power source	string	optional

6.1 Example DataLogger JSON String

```
{
 "manufacturer_s": "MT ‘r Us",
 "model_s": "Broadband 2000",
 "serial_s": "0128947850230",
 "notes_s": "Intern dropped the data logger on a shovel.",
 "timing_system.type_s": "GPS",
 "timing_system.drift_d": 0,
 "timing_system.uncertainty_d": .0000016,
 "timing_system.notes_s": "only works when sky is clear",
 "firmware.version_s": "1.0",
 "firmware.date_s": "2020-01-01",
 "firmware.author_s": "R. Phase",
 "n_channels_i": 5,
 "n_channels_used_s": 4,
 "power_source.type_s": "solar panel and battery]",
 "power_source.voltage.start_d": 13.1,
 "power_source.voltage.end_d": 12.0,
 "power_source.notes_s": "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 6: Attributes for Electric category

Metadata Key	**Description**	**Type**	**Required**
dipole_length_d	length of dipole [m]	float	compulsory
channel_number_i	channel number [1 | 2 | 3 | 4 | 5 | 6 |...]	int	compulsory
component_s	[Ex | Ey | Ez]	string	compulsory
azimuth_d	azimuth of dipole N = 0, E = 90 [degrees]	float	compulsory
positive.ID_s	sensor id number	string	compulsory
positive.latitude_d	positive sensor location latitude [degrees (hh.mmss)]	float	optional
positive.longitude_d	positive sensor location longitude [degrees (hh.mmss)]	float	optional
positive.elevation_d	positive sensor location elevation [m]	float	optional
positive.datum_s	"positive datum for x, y, z location [WGS84]"	string	optional
positive.type_s	type of electric sensor [Ag-AgCl | Pb-PbCl | ...]	string	compulsory
positive.manufacturer_s	electric sensor manufacturer	string	compulsory
positive.notes_s	notes on electric sensor	string	optional
negative.ID_s	sensor id number	string	compulsory
negative.longitude_d	negative sensor location latitude [degrees (hh.mmss)]	float	optional
negative.latitude_d	negative sensor location longitude [degrees (hh.mmss)]	float	optional
negative.elevation_d	negative sensor location elevation [m]	float	optional
negative.datum_s	"negative datum for x, y, z location [WGS84]"	string	optional
negative.type_s	type of electric sensor [Ag-AgCl | Pb-PbCl | ...]	string	compulsory
negative.manufacturer_s	electric sensor manufacturer	string	compulsory
negative.notes_s	notes on electric sensor	string	optional
contact_resistance.start_A_d	contact resistance at beginning of measurement, positive polarity [Ohm]"	float	optional
contact_resistance.start_B_d	contact resistance at beginning of measurement, negative polarity [Ohm]	float	optional
contact_resistance.end_A_d	contact resistance at end of measurement, positive polarity [Ohm]	float	optional
contact_resistance.end_B_d	contact resistance at end of measurement, negative polarity [Ohm]	float	optional
ac.start_d	AC at start of measurement [V]	float	optional
ac.end_d	AC at end of measurement [V]	float	optional
dc.start_d	DC at start of measurement [V]	float	optional
dc.end_d	DC at end of measurement [V]	float	optional

Table 7: Attributes for Electric category continued

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

7.1 Example Electric Channel JSON String


```
{
 "dipole_length_d": 59.7,
 "channel_number_i": "1",
 "component_s": "EX",
 "azimuth_d": 0,
 "positive.id_s": "101",
 "positive.latitude_d": 35.5578,
 "positive.longitude_d": -117.38754,
 "positive.elevation_d": 103.4,
 "positive.datum_s": "WGS84",
 "positive.type_s": "Ag-AgCl",
 "positive.manufacturer_s": "Zaps",
 "positive.notes_s": "Sitting on the shelf since last year",
 "negative.id_s": "102",
 "negative.latitude_d": 35.5588,
 "negative.longitude_d": -117.38754,
 "negative.elevation_d": 105.8,
 "negative.datum_s": "WGS84",
 "negative.type_s": "Ag-AgCl"
 "negative.manufacturer_s": "Zaps",
 "negative.notes_s": "Sitting on the shelf since last year",
 "contact_resistance.start_A_d": 1200.0,
 "contact_resistance.start_B_d": 1210.0,
 "contact_resistance.end_A_d": 1205.0,
 "contact_resistance.end_B_d": 1205.0,
 "ac.start_d": 0.03,
 "ac.end_d": 0.04,
 "dc.start_d": 0.001,
 "dc.end_d": 0.002,
 "units_s": "counts",
 "sample_rate_d": 256,
 "notes_s": "cables chewed on 2020-01-07",
 "data_quality.rating_d": 3,
 "data_quality.warning_notes_s": "cables chewed 2020-01-07",
 "data_quality.warning_flags_s": "Nan",
 "data_quality.author_s": "Q. Sea",
 "filter.name_s": "[counts2mv, datalogger024]",
 "filter.notes_s": "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 8: Attributes for Magnetic category

Metadata Key	**Description**	**Type**	**Required**
sensor.type_s	type of magnetic sensor [Induction Coil | flux gate | ...]	string	compulsory
sensor.manufacturer_s	magnetic sensor manufacturer	string	compulsory
sensor.notes_s	notes on sensor	string	compulsory
sensor.ID_s	sensor id number	string	compulsory
channel_number_i	channel number [1 | 2 | 3 | 4 | 5 | 6 |...]	int	compulsory
component_s	[Hx | Hy | Hz]	string	compulsory
azimuth_d	azimuth in `station\_coordinates_s` [degrees]	float	compulsory
longitude_d	sensor longitude degrees	float	compulsory
latitude_d	sensor latitude in degrees	float	compulsory
elevation_d	sensor elevation in meters	float	compulsory
datum_s	datum for location [WGS84 | ...]	string	compulsory
units_s	units of magnetic field data [counts | mV | ...]	string	compulsory
sample_rate_d	sample rate of magnetic channel (samples.second)	float	compulsory
h_field_min.start_d	minimum h-field value at beginning of measurement	float	optional
h_field_max.start_d	maximum h-field value at beginning of measurement	float	optional
h_field_min.end_d	minimum h-field value at end of measurement	float	optional
h_field_max.end_d	maximum h-field value at end of measurement	float	optional
h_field.units_s	units of h-field measurement [nT | ...]	string	optional
notes_s	notes on magnetic field measurments	string	optional
data_quality.rating_d	data quality rating based on some sort of statistic	float	optional
data_quality.warning_notes_s	any warnings about data quality	string	optional
data_quality.warning_flags_s	a value flagging bad data	float	optional
data_quality.author_s	person who did QC.QA on data	string	optional
filter.name_s	filter name in filter table, can be a list. Needs to be ordered in which filters were applied	string	optional
filter.notes_s	any notes on the filtering	string	optional
filter.applied_b	have filters been applied [True | False]	string	compulsory

8.1 Example Magnetic Channel JSON String


```
{
 "sensor.type_s": "Induction Coil",
 "sensor.manufacturer_s": "MT ‘r Us",
 "sensor.notes_s": "new coil",
 "sensor.id_s": "2149",
 "channel_number_i": 5,
 "component_s": "HZ",
 "azimuth_d": 90,
 "longitude_d": -117.0,
 "latitude_d": 45.0,
 "elevation_d": 107.4,
 "datum_s": "WGS84",
 "units_s": "counts",
 "sample_rate_d": 256,
 "h_field_min.start_d": -10,
 "h_field_max.start_d": 10,
 "h_field_min.end_d": -9,
 "h_field_max.end_d": 9,
 "h_field.units_s": "nT",
 "notes_s": "not buried all the way ",
 "data_quality.rating_d": 4,
 "data_quality.warning_notes_s": "windy during the day",
 "data_quality.warning_flags_s": "Nan",
 "data_quality.author_s": "Q. Sea",
 "filter.name_s": "[counts2mv, datalogger024, coil2149]",
 "filter.notes_s": "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_s` and formatted according to `type_s`. 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_s: "[counts2mv, notch60hz, e_gain]"` and `applied_b: "[true, false, true]"`.

Table 9: Attributes for Filters

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

9.1 Example Filter JSON String

```
{
 "type_s": "look up",
 "name_s": "coil_8897",
 "units_in_s": "mV",
 "units_out_s": "mV",
 "calibration_date_s": "2015-07-01",
 "notes_s": "interpolated from poles and zeros"
}
```

10 Auxiliary Channels

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

Table 10: Attributes for Auxiliary category

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

10.1 Example Auxiliary JSON String

```
{
"type_s": "temperature",
"units_s": "celsius",
"channel_number_i": "6",
"component": "None",
"sample_rate_d": "256",
"notes_s": "internal data logger temperature"
"data_quality.rating_d": 4,
"data_quality.warning_notes_s": "windy during the day",
"data_quality.warning_flags_s": "Nan",
"data_quality.author_s": "Q. Sea",
"filter.name_s": "[counts2c]"
"filter.notes_s": "Calibrated 2018-01-01",
"filter.applied_b": "true"
}
```