CHIME Archive Data Format

Version 4.3.0

Adam Hincks, Kiyoshi Masui, Richard Shaw, Seth Siegel, Don Wiebe
for
The CHIME Collaboration

March 8, 2021

Contents

0 Updating This Document

The TeX source for this document can be obtained from the `ch_memos` repository on GitHub:

> https://github.com/chime-experiment/ch_memos

in the subdirectory `/acq_format/`.

Every time a new version of this document is published, please remember to increment the archive version number, and describe the changes in Table 3 below. We use semantic versioning for the version number:

- If you make an *incompatible* change (one that renders old code unable to read new data of a type it was able to read before the change), **then** increment the first number (MAJOR) and reset the other two numbers to zero;
- If you add a *new data format* (for which no reader code exists) or make a *compatible* change to an existing data format (so that old code can still read new data as it could before), **then** leave the first number as-is, increment the second number (MINOR) and reset the third number to zero;
- If you make other changes to this document, including fixing corriganda and errata, which do not result in changes to the data formats per se, **then** leave the first two numbers as-is, and increment the third number (PATCH).

Also update the current version number in the title.

To publish the updated document, add it as a new revision of Document #0005 in the CHIME Document Library:

> https://bao.chimenet.ca/doc/documents/5

1 Definitions

acquisition – An uninterrupted collection of data with constant correlator parameters. There are several *types* of acquisition, listed in §3. An acquisition consists of one or more chunks all collected in one directory.

chunk – A file or group of files corresponding to a chunk of time within an acquisition. There is no gap in time between chunks.

data file – A single file of data within an acquisition. If the file contains all the frequency channels, then it alone constitutes a chunk. If frequencies are spread across multiple files, then that group of files constitutes a chunk.

2 Acquisition Naming

The directory that constitutes an acquisition has the format:

```
{YYYYMMDD}T{HHMMSS}Z_{instrument-name}_{type}
```

The characters in brackets are variable:

- `YYYYMMDD` – The year (YYYY), month (MM) and day (DD) at which acquisition began.

- HHMMSS – The hour (HH), minute (MM) and second (SS), in 24-hour format, at which acquisition began. The date and the time are UTC (i.e., ≈ Greenwich Mean Time).
- instrument-name – A short, memorable name for the instrument.
  - *Correlator / Raw ADC / Digital Gain / Gain / Flag Input Data:* The instrument name for correlators is a hash for a specific correlator, where a correlator is a unique combination of ADC, FPGA and GPU units. (In early pathfinder, a correlator is simply a single ADC + FPGA unit.) See Table 1 for a list of current correlator hashes.
  - *Pathfinder Housekeeping Data:* The instrument name for the Pathfinder housekeeping system is ben.
  - *Prometheus Data:* The instrument name for the exported Prometheus data is simply chime.
  - *Weather Data:* The instrument name for the new, multi-station data format is simply chime. The instrument for the obsolete, single-station data format was mingun.
  - *Absorber Data:* The instrument name for the absorber data is chime.
- type – One of corr (§3.1), rawadc (§3.2), digitalgain (§3.3), gain (§3.4), flaginput (§3.5), hk (§3.6), hkp (§3.7), weather (§3.8), misc (§3.9), hfb (§3.10).

For example, 20140130T002246Z_abbot_corr is an acquisition of correlator data that began at 00:22:46 UTC on 30 January 2014 on the correlator named ‘abbot’.

3 Acquisition Types and Files

Data files live in acquisition directories (§2). For any given acquisition type, there can be several types of files. The following are the types of acquisitions and which files they contain.

3.1 Correlator Acquisitions

Correlator acquisitions are the main science product of CHIME. They contain two kinds of files.

Correlator Hash	Description
abbot	Eight-channel, FPGA-only correlator. Serial number: 29821-0000-0033
stone	Eight-channel, FPGA-only correlator. Serial number: 29821-0000-0003
vincente	Eight-channel, FPGA-only correlator. Serial number: 29821-0000-0028
blanchard	Sixteen-channel FPGA + GPU correlator. FPGA serial number: ?.
first9ucrate	Sixteen-channel FPGA + GPU correlator.
slotXX	One slot in the FPGA crate operated as a sixteen-channel correlator with the GPU.

Table 1: List of correlator hashes.

3.1.1 Correlator Data Files

Contents – Visibilities.
Naming – {NNNNNNNN}_{FFFF}.h5

- {NNNNNNNN} – An eight-digit integer indicating the start time of the chunk (§1) in seconds since the beginning of the acquisition.
- {FFFF} – A four-digit integer for indexing the division of frequencies between multiple files in a single chunk (§1); if all frequencies are in a single file, then this will simply be 0000.

File type – HDF5
Datasets – vis (§4.1.1.1), gain_coeff (§4.1.1.5), gain_exp (§4.1.1.6), fpga_hk (§4.1.1.3), gate{N}_diff (optional; §4.1.1.2)
Flags – vis_weight (§4.1.2.1), missing_freq (§4.1.2.2), misc (§4.1.2.3), fpga_count (§4.1.2.4), lost_packet_count (§4.1.2.5), rfi_count (§4.1.2.6)

3.1.2 Pathfinder Correlator Log Files

Contents – Run-time log information from ch_master.
Naming – ch_master.log
File type – plain text

3.2 Raw ADC Aquisitions

Raw ADC snapshots from the FPGA's. 1-4 consecutive frames of 8 bit adc values from each analog input possible.

3.2.1 Raw Data Files

Contents – Raw ADC data from the FPGA's.
Naming – {NNNNNN}.h5

- {NNNNNN} – A six-digit integer counting files since the beginning of the acquisition.

File type – HDF5
Datasets – timestamp (fpga_count, ctime), slot (slot number), crate (crate number), adc_input (ADC input number), timestream (8-bit timestream snapshot data)

3.3 Digital Gain Acquisitions

Digital gain acquisitions are produced by ch_master. They contain the digital gains that the FPGAs applied to the channelized data.

3.3.1 Digital Gain Data Files

Contents – Digitals gains.
Naming – `{NNNNNNNN}.h5`

- `{NNNNNNNN}` – An eight-digit integer indicating the start time of the chunk (§1) in seconds since the beginning of the acquisition.

File type – HDF5
Datasets – `update_id` (§4.1.1.4), `gain_coeff` (§4.1.1.5), `gain_exp` (§4.1.1.6), `compute_time` (§4.1.1.7)

3.4 Gain Acquisitions

Gain acquisitions are produced by the calibration broker. They contain the complex gain for each correlator input and frequency as a function of time, and can be used to calibrate the visibility data.

3.4.1 Gain Data Files

Contents – Complex gains.
Naming – `{NNNNNNNN}.h5`

- `{NNNNNNNN}` – An eight-digit integer indicating the start time of the chunk (§1) in seconds since the beginning of the acquisition.

File type – HDF5
Datasets – `update_id` (§4.1.1.4), `source_gains` (§4.1.1.8), `gain` (§4.1.1.9), `source_weights` (§4.1.1.10), `weight` (§4.1.1.11)

3.5 Flag Input Acquisitions

Flag input acquisitions are produced by the flagging broker. They contain flags indicating which correlator inputs are considered good as a function of time.

3.5.1 Flag Input Data Files

Contents – Input flags.
Naming – `{NNNNNNNN}.h5`

- `{NNNNNNNN}` – An eight-digit integer indicating the start time of the chunk (§1) in seconds since the beginning of the acquisition.

File type – HDF5
Datasets – `update_id` (§4.1.1.4), `source_flags` (§4.1.1.12), `flag` (§4.1.1.13)

3.6 Pathfinder Housekeeping Acquisitions

Housekeeping acquisitions use ATMEL analogue-to-digital boards and currently monitor LNA and FLA temperatures, and optionally accelerometers. They contain three types of file, as follows.

3.6.1 Housekeeping Data Files

Contents – Housekeeping data.
Naming – `{subsys}_{NNNNNNNN}.h5`

- `{subsys}` – A single acquisition can contain data from multiple ATMEL boards; each board constitutes a 'subsystem' and has its own series of HDF5 files. Current possible values of `{subsys}` are `lna`, `fla`, `accel`. Subsystem names are listed in the ATMEL ID file (§3.6.2).
- `{NNNNNNNN}` – An eight-digit integer indicating the start time of the chunk (§1) in seconds since the beginning of the acquisition.

File Type – HDF5
Datasets – Either `data` (§4.1.1.14) or `mux{NN}` (§4.1.1.15)
Flags – *None*

3.6.2 Pathfinder Housekeeping ATMEL ID Files

Contents – A list of `subsys` names (c.f. §3.6.1) together with the serial IDs of their respective ATMEL boards.
Naming – `atmel_id.dat`
File Type – Plain text

3.6.3 Pathfinder Housekeeping Log Files

Contents – Run-time log information from `ch_hk`.
Naming – `ch_hk.log`
File type – plain text

3.7 Prometheus Housekeeping Acquisitions

This section needs to be written!

3.8 Weather Acquisitions

A weather acquisition may either be a single-station acquisition, or else a multi-station acquistion. Single-station and multi-station data files are never both present in the same acqusition.

3.8.1 Single-station Weather Data Files

Contents – Weather data from a single weather station.

Naming – `{YYYYMMDD}.h5`

- `{YYYYMMDD}` – The year-month-day of the weather data; data always run from midnight to midnight. Files for 2017-04-24 and earlier use UTC midnight. Files for 2017-04-25 and later use local midnight.

File type – HDF5

Datasets – `barometer`, `pressure`, `altimeter`, `inTemp`, `outTemp`, `inHumidity`, `outHumidity`, `windSpeed`, `windDir`, `windGust`, `windGustDir`, `rainRate`, `rain`, `dewpoint`, `windchill`, `heatindex` (§4.1.1.16)

3.8.2 Multi-station Weather Data Files

Contents – Weather data from (potentially) multiple weather stations.

Naming – `{YYYYMMDD}.h5`

- `{YYYYMMDD}` – The year-month-day of the weather data; data always run from midnight to midnight. These are always UTC times.

File type – HDF5

Groups – Data from each weather station is in a separate group whose name is the station name. Each group contains the following datasets:

Datasets – `barometer`, `pressure`, `altimeter`, `inTemp`, `outTemp`, `inHumidity`, `outHumidity`, `windSpeed`, `windDir`, `windGust`, `windGustDir`, `rainRate`, `rain`, `dewpoint`, `windchill`, `heatindex` (§4.1.1.16)

3.9 Miscellaneous Acquisitions

Miscellaneous acquisitions are used to store non-HDF5 data created irregularly for various puropses. A miscellaneous acqusition contains one or more miscellaneous data files, as specified below, which are `tar` archives, optionally compressed with `gzip`, `bzip2`, or `xz`, containing the miscellaneous data themselves.

3.9.1 Miscellaneous Data Files

Contents – Miscellaneous data files may contain any allowed `tar` archive member, typically regular files and directories.

Naming – `{NNNNNNNNN}_{TYPE}.misc.tar[.gz|.bzip2|.xz]`

- `{NNNNNNNN}` – An eight-digit serial number. A specific interpretation of the serial number is not specified, but the intention is that it be used for date, time, time-offset, &c., when appropriate. Multiple miscellaneous data files in the same acqusition are ordered by increasing serial number.
- `{TYPE}` – The miscellaneous data type (`rfi`, `pulsar`, &c.). Recorded in the data index for searchability. If two miscellaneous data files in the same acquisition have different types, they may have the same serial number.
- `[.gz|.bzip2|.xz]` – An optional compression suffix, for compressed archives. Although common for compressed archives, this should *never* be combined with the `.tar` part of the filename to produce, say, `.tgz`.

File type – `tar` archive, optionally compressed.

Metadata – All miscellaneous data files may contain an archive member called `METADATA.json`. This is a JSON file containing an object providing additional metadata for this miscellaneous data file. If the miscellaneous data are temporal in nature, this object should contain at least the members `start_time` and `finish_time` with ISO 8601-formatted date-times (`YYYY-MM-DDTHH:MM:SSZ`) indicating the time interval for the file. If multiple archive members are called `METADATA.json`, only the last one is considered.

3.10 Absorber Acquisitions

Absorber acquisitions are produced by the correlator. They contain 10s integrations of FRB beams at full spectral resolution for each CHIME frequency.

Contents – Absorber data from the correlator

Naming – `hfb_{NNNNNNNN}_{FFFF}.h5`

- `{NNNNNNNN}` – An eight-digit integer indicating the start time of the chunk (§1) in seconds since the beginning of the acquisition.
- `{FFFF}` – A four-digit integer for indexing the division of frequencies between multiple files in a single chunk (§1); if all frequencies are in a single file, then this will simply be `0000`.

File type – HDF5

Datasets – `hfb` (§4.1.1.17)

Flags – `hfb_weight` (§4.1.2.7), `frac_lost` (§4.1.2.8)

4 HDF5 Format

With the exception of miscellaneous data (§3.9), all other acquisition types (c.f., §3) use HDF5 for their data files. While the content of their data differ significantly, all of them have the same basic structure and use the same semantics.

4.1 Datasets

Each dataset consists of an array of data, and must have an attribute `axis` which is an array, of the same rank as the data array, that names each axis. For example, the dataset `vis` has

```
        axis = {‘freq’, ‘prod’, ‘time’}
```

where each of these labels corresponds to an attribute of the `index_map` group that interprets the indices of that axis (see §4.2).

A dataset may also, optionally, have an attribute `axis_rate`, also of the same rank as the data array, that indicates how many records exist per axis index. For example, if a housekeeping dataset `mux00` had `axis = {‘time’}` and `axis_rate = {5}`, then there would be five records in the dataset for each entry in the `index_map/time` dataset. In short, using the `axis_rate` allows datasets to share an index map but run at different rates.

The datasets follow, and the type of acquisition to which they belong are indicated therein—not all of these datasets will occur in all types of acquisition. All array definitions are in `C` order, i.e., the final axis is the fastest-varying.

4.1.1 Data

These are datasets which contain actual science data.

4.1.1.1 `vis`

Contents – The visibility products from the correlators.
Acquisition Type – Correlator (§3.1)
Axes – {`‘freq’`, `‘prod’`, `‘time’`} (§§4.2.2, 4.2.1, 4.2.5.1)
Data Type – Compound of two 32-bit integers (one for each of the real and imaginary parts).
Data Size – 64 bits (32 bits for each of the real and imaginary parts).

4.1.1.2 `gated_vis{N}`

Contents – If there are gated data, a weighted sum of the gates are stored in this dataset. There can be multiple gated datasets with different periods, integration times, weights, and so {N} $= 0, 1, 2, \ldots$ indicates which gated data this dataset refers to. The intension is to have one such dataset per transit/observation and each dataset may begin and end within any chunk of an acquisition. The identifier {N} increments with each new observation and is never reused within an acquisition, even if its last use was in a long-passed chunk.

Acquisition Type – Correlator (§3.1)

Axes – {`'freq'`, `'prod'`, `'gated_time{N}'`} (§§4.2.2, 4.2.1, 4.2.6)

Data Type – Compound of two 32-bit integers (one for each of the real and imaginary parts).

Data Size – 64 bits (32 bits for each of the real and imaginary parts).

Dataset Attributes – There are four required attributes on the dataset:
`folding_period` – the folding period in seconds
`folding_start` – a reference time for the folding start
`gate_weight` – an array of double precision floats giving the weight to each gate which is summed in this dataset. The length of this array is equal to the number of gates.
`description` – a short description of what the gating was for e.g. 'observation of PSR B0329+54'

4.1.1.3 `fpga_hk`

Contents – System-monitoring information, such as temperatures and voltage levels, from the FPGA.

Acquisition Type – Correlator (§3.1)

Axes – {`'fpga'`, `'fpga_hk'`, `'time'`} (§§4.2.7, 4.2.5.2)

Data Type – Float.

Data Size – 16 bits.

4.1.1.4 `update_id`

Contents – Unique identifier of each dynamic update to the real-time pipeline.

Naming – `{update-type}_{YYYYMMDD}T{HHMMSS}.{FFFFFF}Z_{notes}`

- `update-type` – One of `gain` or `flaginput`.
- `YYYYMMDD` – The year (YYYY), month (MM) and day (DD) at which the update was created.
- `HHMMSS` – The hour (HH), minute (MM) and second (SS), in 24-hour format, at which the update was created. The date and the time are UTC (i.e., ≈ Greenwich Mean Time).
- `FFFFFF` – The fractional second at which the update was created.
- `notes` – Supplementary information about the update.

Acquisition Type – Digital Gain (§3.3), Gain (§3.4), Flag Input (§3.5)

Axes – {`'update_time'`} (§4.2.9)

Data Type – String.

Data Size – 64 bits.

4.1.1.5 `gain_coeff`

Contents – The coefficient of the digital gain applied to the channelized data from the ADC inputs, such that the digital gain is `gain_coeff` $\times\ 2^{\texttt{gain\_exp}}$.
Acquisition Type – Correlator (§3.1), Digital Gain (§3.3)
Axes – *Correlator:* {`‘freq’, ‘input’, ‘time’`} (§§4.2.2, 4.2.1, 4.2.5.1)
Digital Gain: {`‘update_time’, ‘freq’, ‘input’`} (§§4.2.9, 4.2.2, 4.2.1)
Data Type – Compound of two 32-bit integers (one for each of the real and imaginary parts).
Data Size – 64 bits.

4.1.1.6 `gain_exp`

Contents – The exponent of the digital gain applied to the channelized data from the ADC inputs, such that the digital gain is `gain_coeff` $\times\ 2^{\texttt{gain\_exp}}$. All frequencies of a given input share the same exponent.
Acquisition Type – Correlator (§3.1), Digital Gain (§3.3)
Axes – *Correlator:* {`‘input’, ‘time’`} (§§4.2.1, 4.2.5.1)
Digital Gain: {`‘update_time’, ‘input’`} (§§4.2.9, 4.2.1)
Data Type – Ingeger. (Signed?)
Data Size – 32 bits.

4.1.1.7 `compute_time`

Contents – C-time at which the digital gains were computed for each ADC input.
Acquisition Type – Digital Gain (§3.3)
Axes – {`‘update_time’, ‘input’`} (§§4.2.9, 4.2.1)
Data Type – Float.
Data Size – 32 bits.

4.1.1.8 `source_gains`

Contents – Complex gain inferred from different sources.
Acquisition Type – Gain (§3.4)
Axes – {`‘update_time’, ‘source’, ‘freq’, ‘input’`} (§§4.2.9, 4.2.10, 4.2.2, 4.2.1)
Data Type – Complex. (Float?)
Data Size – 64 bits.

4.1.1.9 `gain`

Contents – Best estimate of the complex gain, obtained by taking the product of the `source_gains` (§4.1.1.8) along the `source` axis.
Acquisition Type – Correlator (§3.1), Gain (§3.4)
Axes – {`‘update_time’, ‘freq’, ‘input’`} (§§4.2.9, 4.2.2, 4.2.1)
Data Type – Complex. (Float?)
Data Size – 64 bits.

4.1.1.10 `source_weights`

Contents – $1/\sigma^2$ where σ indicates the uncertainty on the complex gain obtained from differrent sources.
Acquisition Type – Gain (§3.4)
Axes – {`‘update_time’, ‘source’, ‘freq’, ‘input’`} (§§4.2.9, 4.2.10, 4.2.2, 4.2.1)
Data Type – Float.
Data Size – 32 bits.

4.1.1.11 `weight`

Contents – $1/\sigma^2$ where σ indicates the uncertainty on the complex gain obtained by propagating the uncertainty in the `source_weights` (§4.1.1.10).
Acquisition Type – Gain (§3.4)
Axes – {`‘update_time’, ‘freq’, ‘input’`} (§§4.2.9, 4.2.2, 4.2.1)
Data Type – Float.
Data Size – 32 bits.

4.1.1.12 `source_flags`

Contents – Flag indicating the good inputs inferred from different sources.
Acquisition Type – Flag Input (§3.5)
Axes – {`‘update_time’, ‘source’, ‘input’`} (§§4.2.9, 4.2.10, 4.2.1)
Data Type – Boolean.
Data Size – 1 bit.

4.1.1.13 `flag`

Contents – Best estimate of the good input flag, obtained by taking the logical AND of some subset of the `source_flags` (§4.1.1.12) along the `source` axis.
Acquisition Type – Flag Input (§3.5)
Axes – {`‘update_time’, ‘input’`} (§§4.2.9, 4.2.1)
Data Type – Boolean.
Data Size – 1 bit.

4.1.1.14 `data`

Contents – Housekeeping values for subsystems (§3.6.1) with no multiplexing.
Acquisition Type – Housekeeping (§3.6)
Axes – {`‘time’, ‘data_chan‘`} (§§4.2.5.1, 4.2.8)
Data Type – Float.
Data Size – 32 bits.

4.1.1.15 `mux{NN}`

Contents – Housekeeping values for subsystems (§3.6.1) with multiplexing; `{NN}` = $0, 1, 2, \ldots, 7$ refers to which multiplex channel this dataset records.
Acquisition Type – Housekeeping (§3.6)
Axes – {`‘time’`, `‘mux{NN}_chan‘`} (§§4.2.5.1, 4.2.8)
Data Type – Float.
Data Size – 32 bits.
Notes – This kind of dataset includes the attribute `mux_address` which records which multiplexer `NN` the data came from.

4.1.1.16 Weather Datasets

This section describes all of the following datasets: `barometer`, `pressure`, `altimeter`, `inTemp`, `outTemp`, `inHumidity`, `outHumidity`, `windSpeed`, `windDir`, `windGust`, `windGustDir`, `rainRate`, `rain`, `dewpoint`, `windchill`, `heatindex`:

Contents – Weather data from a single weather station.
Acquisition Type – Weather (§3.8)
Axes – *For single-station weather data (§3.8.1)*: {`‘time’`} (§4.2.5.2)
For multi-station weather data (§3.8.2): {`‘station_time_{S}’`} (§4.2.11)
Data Type – Float. (*Note:* Although data are stored as IEEE floats, the data are received by `wview` as 15.16 fixed-point values.)
Data Size – 32 bits.
Notes – Each of these datasets includes the attribute `units` which gives the physical units of its data. The names of the datasets are exactly the same as in the `wview` database from which the data are scraped.

4.1.1.17 `hfb`

Contents – Absorber data from the correlators.
Acquisition Type – Absorber (§3.10)
Axes – {`‘freq’`, `‘subfreq’`, `‘beam’`, `‘time’`} (§§4.2.2, 4.2.3, 4.2.4, 4.2.5.1)
Data Type – 32-bit floats.
Data Size – 32 bits.

4.1.2 Flags

Flags are stored as datasets in the `flags` group. Defined flags are as follows.

4.1.2.1 `vis_weight`

Contents – Proportional to total integration time for sample, accounting for all flags.
Acquisition Type – Correlator Data
Axes – {`‘freq’, ‘prod’, ‘time’`} (§§4.2.2, 4.2.1, 4.2.5.1)
Data Type – Unsigned ingeger.
Data Size – 8 bits.

4.1.2.2 `missing_freq`

Contents – Bitfield indicating if a time slice has any missing frequencies due to broken GPU connexions (0 indicates a missing frequency).
Acquisition Type – Correlator Data
Axes – {`‘time’`} (§4.2.5.1)
Data Type – bitfield, using an array of `uint8_t`; the zeroth frequency bin (as recorded in the `freq` index map; §4.2.2) is represented by the LSB in the array
Data Size – N bits, where N is the number of frequency bins rounded up to the nearest multiple of 8

4.1.2.3 `misc`

Contents – Miscellaneous flags
Acquisition Type – Correlator (all?) Data
Axes – {`‘time’`} (§4.2.5.1)
Data Type – Bitfield stored as array of `uint8_t`.
Data Size – 128 bits.

The meaning of each of the 128 possible flags will be stored in the header as they are allocated.

4.1.2.4 `fpga_count`

Contents – Number of flag trips coming from the FPGA
Acquisition Type – Correlator Data
Axes – {`‘freq’, ‘input’, ‘time’`} (§§4.2.2, 4.2.1, 4.2.5.1)
Data Type – Something like `struct { uint32_t adc; uint32_t fft; uint32_t scalar; }`.
Data Size – 96 bits.

This is not enough information to reproduce the integration time, but gives detailed statistics on how often we overflow.

4.1.2.5 `lost_packet_count`

Contents – Number of packets dropped between FPGA and GPUs
Acquisition Type – Correlator Data
Axes – {`'freq'`, `'time'`} (§§4.2.2, 4.2.5.1)
Data Type – Unsigned Integer.
Data Size – 32 bits.

4.1.2.6 `rfi_count`

Contents – Number of high-cadence samples discarded due to RFI
Acquisition Type – Correlator Data
Axes – {`'freq'`, `'time'`} (§§4.2.2, 4.2.5.1)
Data Type – Unsigned Integer.
Data Size – 32 bits.

4.1.2.7 `hfb_weight`

Contents – Proportional to total integration time for sample, accounting for all flags.
Acquisition Type – Absorber Data
Axes – {`'freq'`, `'sub-freq'`, `'beam'`, `'time'`} (§§4.2.2, 4.2.3, 4.2.4, 4.2.5.1)
Data Type – Float.
Data Size – 32 bits.

4.1.2.8 `frac_lost`

Contents – Fraction of samples lost in integration.
Acquisition Type – Absorber Data
Axes – {`'freq'`, `'time'`} (§§4.2.2, 4.2.5.1)
Data Type – Float.
Data Size – 32 bits.

4.2 Index Maps

Data are stored in multi-dimensional arrays (§4.1), the indices of which refer to some physical quantity. For example, correlator data are stored in a three-dimensional array (see §4.1.1.1), with one index for each of visibility channel, frequency channel and time. A map is needed for each of these dimensions, to convert each index to visibility product, frequency centre and width, and time of day, respectively. See Table 2 for an example of such a map.

Each dataset, subsequently has an attribute called '`axis`', which is an array of strings the same rank as the dataset. The string refers to the name of a dataset in the `index_map` group. For example, the `axis` attribute of the visibility dataset would be:

```
{'freq', 'prod', 'time'}
```

The index maps follow.

4.2.1 prod and input

Visibility products are defined by a pair of inputs. An ADC input is described by a unique integer determined by its location in the FPGA rack known as the *channel ID*. It is defined in CHIME Document #0165.

Now, in the layout database, these correlator inputs are defined as:

```
{crate_sn}{r}{c}
```

where `r` is the slot number in the ADC/FPGA crate and `c` is the SMA number (bottom-to-top) on the card and `crate_sn` is the serial number of the ADC crate. For example, `K7BP16-00041201` denotes SMA plug 1 of slot 12 in the crate `K7BP16-0004`.

In the HDF5 file, the channel ID mapping to correlator input is recorded in the `index_map/input` map:

```
struct input_t {
  int chan_id;
  char *correlator_input;
};
```

A visibility is then the product of two inputs:

```
struct prod_t {
  int input_a;
  int input_b;
};
```

where `input_a` and `input_b` are (implicit) indices of the `index_map/input` table.

See Table 2 for an example of `index_map/prod` and `index_map/input`.

4.2.2 freq

A frequency channel is defined by a frequency centre and width (in megahertz), and is represented as a struct in `C` by:

```
struct freq_chan_t {
  double centre;
  double width;
};
```

The `index_map/freq` dataset is an array mapping indice of the frequency dimension of the visibility dataset to such pairs.

index_map/prod

	input_a	input_b
0	0	0
1	0	1
2	0	2
3	0	3
...	...	...
31	0	31
32	1	1
33	1	2
...	...	...
526	30	31
527	31	31

index_map/input

	chan_id	correlator_input
0	32	K7BP16-00041600
1	33	K7BP16-00041601
...	...	...
15	47	K7BP16-00041615
16	96	K7BP16-00041500
17	97	K7BP16-00041501
...	...	...
31	111	K7BP16-00041515

Table 2: Example showing the structure of the `index_map/prod` and `index_map/input` datasets. In this example, two slots are used for a total of 32 inputs, and for these slots the channel ID's do not start at 0.

4.2.3 `subfreq`

The `index_map/subfreq` dataset is an array mapping the order of the sub-frequencies in the main frequency dimension, which is determined by the Nyquist zone. Each value gives the difference of the sub-frequency from the central frequency.

4.2.4 `beam`

The `index_map/beam` dataset is an array mapping index of the beams. The position of the beams on the sky is defined in: https://chimefrb.github.io/frb_common/build/html/beam_model.html.

4.2.5 `time`

4.2.5.1 Visibilities

The time indices of the visibility data are mapped to the fractional C-time (i.e., having sub-second resolution) at the *beginning* of the time-sample in the datasets `index_map/time`; additionally, the FPGA counter is also recorded. The structure is:

```
struct timestamp_t {
  unsigned int fpga_count;
  double ctime;
};
```

4.2.5.2 Housekeeping and Weather Data

This index map name is also used for various housekeeping and weather data, where it is a simple array of C-times inidicating the time of measurement.

4.2.6 gated_time{N}

A time axis for any gated_vis{N} dataset (§4.1.1.2). This has exactly the same format as the time index map, but gives the timestamp for the corresponding gated dataset. As before the actual name of this will have the {N} replaced with the actual value $\{N\} = 0, 1, 2, \ldots$.

4.2.7 fpga_hk and fpga

In addition to having a time axis, the FPGA housekeeping has an axis denoting which FPGA it was reading from and what housekeeping value it read. They are, respectively, recorded in fpga and fpga_hk.

The index map fpga simply is a list of ADC/FPGA slot serial numbers.

The index map fpga_hk records the name of the housekeeping item and its units (e.g., "core_temp" and "deg C"):

```
struct fpga_hk_field_t {
  char *name;
  char *units;
};
```

4.2.8 data_chan and mux{NN}_chan

Non-multiplexed housekeeping data can contain up to eight channels, which are enumerated in the data_chan index map.

Multiplexed housekeeping data can contain up to sixteeen channels, which are enumerated in the mux{NN}_chan index map, where {NN} indicates the multiplexer.

4.2.9 update_time

The index map update_time indicates the fractional C-time at which an update took effect. The value of a dataset with an update_time axis should be considered a step function in time (i.e., the value at any instant in time is equal to the value at the most recent update time).

4.2.10 source

The index map source is an array of strings, where each element is a short unique identifier of a process that has contributed either input flags or gains.

4.2.11 station_time_{S}

Used in multi-station weather data (§3.8.2), the index map is an array of C-times inidicating the time of observation. These times are *not* guaranteed to be uniformly spaced. The {S} portion is replaced by the name of the weather station reporting (equivalent to the name of the weather station group).

4.3 Calibration Information

Calibration information (currently only for housekeeping instruments) is stored as attributes in a group called ‘/cal’. Each calibration has its own group, specified by a unique name, within ‘/cal’. There are two required attributes for each calibration:

- formula – A string describing how the calibration is to be applied (e.g., ‘a * x + b’).
- units – The string providing physical units of the calibrated data (e.g., ‘mV’).

Any other attributes necessary for defining the calibration should be added *ad hoc*. For example, here’s how a (fictitious) linear calibration from raw units to voltage might be defined:

```
GROUP "cal" {
  GROUP "adc_to_volts" {
    ATTRIBUTE "m": "1.23e4"
    ATTRIBUTE "b": "5.0"
    ATTRIBUTE "formula": "m * x + b"
    ATTRIBUTE "units": "mV"
  }
}
```

The dataset making use of this calibration can then reference it by the name ‘my_linear_cal’.

4.4 Header Attributes

Metadata or ‘header’ information for the HDF5 file are stored as attributes of the base group ‘/’.

4.4.1 Headers in all Acquisition Types

- git_version_tag – For keeping track of which version of code produced an HDF5 file: the output of ‘git describe --tags’.
- system_user – The user on the collection server that initiated the acquisition (normally, root).
- collection_server – The hostname of the collection server.
- instrument_name – The name of instrument (e.g., correlator name) from which the data originate.
- acquisition_name — The name of the acquisition, as per §2.
- archive_version – The archive version number. See Table 3.
  N.B.: Data produced on the collection server, which still needs to be time-transposed, will be prefixed by NT_: e.g., archive_version = NT_2.0.0.
- notes – Any notes entered by a user for this acquisition.

4.4.2 Headers for Correlator Acquisitions

4.4.2.1 FPGA Settings

The settings used by the chFPGA_controller module to configure the FPGA, as returned by the chFPGA_controller.get_config() method, are all included as attributes.

Version	Notes	Date
1.0.0	The earliest archive format. HDF5 files of this version are not labelled as such; any file without a `archive_version` attribute can be assumed to be of this version.	—
2.0.0	Archive format with the `index_map` implemented and transpose of the time-index.	2014-04-30
2.1.0	The `prod` and `input` index maps (§4.2.1) now follow the channel ID system of Doclib #0165.	2014-10-30
2.2.0	Flagging, gating and gains added.	2015-04-02
2.3.0	Weather data and associated datasets added.	2015-04-30
2.4.0	Raw format defined.	2015-06-04
3.2.0	Digital gain, gain, and flag input acquisitions added.	2019-02-08
4.0.0	Multi-station weather data added. Old `raw` format deleted.	2019-12-01
4.1.0	Miscellaneous data added.	2020-03-11
4.3.0	Absorber (HFB) data added.	2021-02-26

Table 3: List of archive versions.

4.4.2.2 Configuration Settings

All of the entries in the `ch_master.conf` file used for an acquisition are stored as attributes. They will be prefixed by a full-stop-separated string of any section names to preserve the hierarchy of `ch_master.conf`.

For example, the following entries in `ch_master.conf`,

```
n_antenna   = 8

[acq]
  base_path          = /data/
  frames_per_file    = 3

  [[udp]]
    port             = 41001
    spf              = 10
```

will spawn the following attributes in the HDF5 file:

```
n_antenna
acq.base_path
acq.frames_per_file
acq.udp.port
acq.udp.spf
```

4.4.2.3 Gating settings

The dataset is required to have a `gated_vis_number` giving the total number of gated datasets in this file. Previous versions of the format are implicitly assumed to have this as `gated_vis_number` = 0.

4.4.3 Headers for RawADC Acquisitions

This section needs to be written!

4.4.4 Headers for Gain Acquisitions

- `version` – Version of the calibration broker used to acquire gains. The calibration broker version number is incremented with any changes to the calibration algorithm.

4.4.5 Headers for Flag Input Acquisitions

- `version` – Version of the flagging broker used to acquire input flags. The flagging broker version is incremented with any changes to the flagging algorithm.
- `combine` – List of the sources that were combined to generate the best estimate of the input flags. The logical AND of the `source_flags` (4.1.1.12) for these source will yield `flag` (4.1.1.13).

4.4.6 Headers for Housekeeping Acquisitions

- `atmel_id` – The serial ID for the ATMEL board used.

4.4.7 Headers for Weather Acquisitions

For single-station weather data (§3.8.1):

- `wview_database` – The path to the database created by `wview` from which the data were scraped.

For multi-station weather data (§3.8.2), there are no additional header attributes, but each weather station group contains the attributes:

- `wview_database` – The path to the database created by `wview` from which the data were scraped.
- `longitude` – The East longitude of the weather station.
- `latitude` – The North latitude of the weather station.
- `description` – A short text description of the station.