

CPD3 Text Based Acquisition System Configuration

Table of Contents

Data Formats	1
Archive Structure	1
Text Syntax	3
Constructed Value Types	12
Examples	65
Minimal Acquisition System	65
Manual Serial Number	65
Manual Metadata	65
Matched Metadata	66
Fixed Control Interface	67
Generic Conversion	68
Simple Scheduling	69
Simple Triggering	70
Carousel Accumulation	71
Generic Input	73
Realtime Export	75
Full System	76
acquisition variable	83
General Settings	83
2B Ozone Monitor Model 205	99
Aerosol Dynamic Magic CPC Model 200	100
Aerodyne CAPS	101
Azonix μ MAC	102
BMI CPC Model 1710	111
BMI CPC Model 1720	112
Brooks PID Controller Model 0254	112
Campbell CR1000 with GMD firmware	118
CSD POPS OPC	127
DMT BCP	128
DMT CCN	130
DMT PAX	136
Ecotech Aurora 3000/4000 Nephelometer	152
Eigenbrodt NMO191 Rain Sampler	179
Generic METAR conversion	180
Generic passive data conversion	182
GMD CLAP-3W or BMI TAP	195
GMD CPC pulse counter box	205
Grimm OPC Models 110x	206

Love PID Controller	207
Magee Aethalometer Model AE33	213
Magee Aethalometer Models AE16/AE21/AE31	220
Magee TCA-08	225
PurpleAir PA-II	226
R.M. Young 86xxx Sonic Anemometer	226
Radiance Research Nephelometer Model M903	227
Radiance Research 1-W PSAP	248
Radiance Research 3-W PSAP	253
Thermo MAAP	261
Thermo Series 49 Ozone Monitor	267
Thermo Series 49iQ Ozone Monitor	279
TSI CPC Models 302x	280
TSI CPC Models 377x	281
TSI CPC Model 3010	282
TSI CPC Model 3781	283
TSI CPC Model 3783	284
TSI Mass Flow Meter Models 4xxx	286
TSI Nephelometer Model 3563	287
Vaisala PWD Models x2	321
Vaisala WMT700 Sonic Anemometer	323
Vaisala WXT Models 5xx	323
Accumulation Controller	325
Generic Command Output	338
Cycle Controller	343
Realtime Data Export Controller	362
Generic Trigger Condition Controller	376

Data Formats

Archive Structure

In CPD3 data, configuration, events and any other time dependent information are all stored as a common form of "data". In most cases the final form of data/configuration is a single time dependent value. How this value is constructed is somewhat complicated but it is analogous to a stack of transparent sheets on top of one another. In this analogy, each sheet has only parts of it filled out allowing for those below it to be seen. That is, the sheets higher in the stack "override" parts of those lower in the stack and/or provide definitions for the parts that were completely "transparent" before. In the simple case of a traditional data time series, the stack is only one deep and contains a single non-transparent real number. For most forms of configuration the stack contains more levels.

Sequence Name

The first level of identification of this data are the components of the total "name" of it. Data within a single name are viewed as representing one property. In the context of traditional data, this means a single parameter in time. For configuration, this is usually a single subsystem or component being configured. The components of the name are as follows.

Station

The station is generally the three letter [GAW station code](#) for the physical site in question. Stations are **not** case sensitive.

An additional station called the "default station" is also present. The default station is assigned the code `_` (a single underscore) and is generally always present in all data. It is most commonly used to provide configuration defaults and can generally be ignored in data querying.

Archive

The archive is the "type" of data being accessed. For example the `raw` archive generally specifies data as it comes out of the acquisition system with no mentor QC applied. Once those data have been passed by the mentor and had corrections applied it enters into the `clean` archive. Passed data averaged to one hour intervals are placed in the `avgh` archive. There are additional special archives that include `configuration` and `events` for the system configuration and event logs, respectively. Archives are **not** case sensitive.

Archives can also have an `_meta` suffix that contains metadata about the data or configuration in the "main" archive. For example the `raw_meta` archive contains the metadata about the `raw` archive. That metadata specifies things like physical units and output formats.

Variable

The variable is what is normally considered as an individual parameter in a time series or a column in a table. For example a single wavelength of scattering would be a single variable. In this context the green scattering from the "S11" nephelometer would be `BsG_S11`.

In archives that are not directly tied to measurements (e.g. the `configuration` archive) the

variable specifies the next level of the hierarchy being addressed. For example `processing` specifies the configuration for parts of the automatic processing system.

Variables **are** case sensitive.

Flavors

The flavors of a parameter are the usually hidden qualifies of it. Parameters may have zero or more flavors applied to them. For example the `pm1` flavor species that the value is size selected to less than 1 μ m diameter. Flavors are **not** case sensitive.

Some common flavors include:

PM1

Data have been size selected to less than 1 μ m diameter.

PM25

Data have been size selected to less than 2.5 μ m diameter.

PM10

Data have been size selected to less than 10 μ m diameter.

stats

This value represents statistics about the average rather than the averaged value itself. For example, this would contain quantiles within the averaging period.

cover

This value contains the coverage fraction (zero to one) of data within the averaging period. For example an hourly average with 45 minutes of valid data would have a coverage of 0.75. When absent, this is assumed to be one (whole time period present). This is normally used to calculate correctly weighted averages to prevent single points from throwing the average off.

Priority

The next component of data values is the concept of priority. The priority is an integer (positive or negative) that defines where in the stack the value should appear. Lower priorities are overridden by higher ones. That is, higher priorities are closer to the "top" of the transparency stack analogy above.

Time Bounds

The final component is the time range each value occupies. That is, every value for both data and configuration in the system has a start and time. Either of these can be "undefined" indicating that the value extends to infinity. Values with infinite starts or ends are considered active by the system for all time up until or only after their finite start or end. If a value has both an undefined start and end time, it is always in effect. In general the stack of values can be thought of as being evaluated at every point in time. That is, layers of the transparency can extend between multiple boundaries and do not necessarily need to align on all levels. This allows parts of the configuration to be defined by different levels corresponding to the times they affect.

Additional Sequences

In general most configuration also contains values in the default station mentioned above. The default stations is designated by a station names just `_` (a single underscore). However, this is not exclusive to configuration. These values form another stack of data placed under the "main" ones in question. That is, values in the default station are always below those from normal stations, but they obey priority sorting within themselves. The values in the default station are used to provide system defaults and templates for more specialized configurations. This allows all stations to share common configuration when possible.

Finally there are values attached to most names in a "metadata" archive. Metadata values are designated by the same name as their corresponding main data values, except that the archive ends with `_meta`. These values do not generally affect the actual values as seen by the system. Instead, they provide additional information about the final values. For traditional data, this can be things like formats, MVCs, wavelengths and descriptions. In the context of configuration, they usually provide information about the possible elements of the configuration or limits on what are acceptable values.

Summary

In summary, data and configuration in CPD3 consist of zero or more layers of individual values. Each value is uniquely defined by several components:

The time range

The start and end time the value is effective for. Outside of this range, the system will not "see" the value at all.

The name

The station, archive, variable and flavors (if any) that determines what the value is actually about.

The priority

An integer specifying the depth in the stack of the value is placed when the final data are calculated.

That is, there can only be a single value for a unique combination of the components. For example, to create multiple levels of configuration active for the same time range they must be given unique priorities.

Text Syntax

Data values in CPD3 can be either simple single datum or complex tree-like structures. That is, a "value" in the context of CPD3 consists of either a map from zero or more "names" to further values or a final single datum. For example a final datum might be a simple real number or a string of text. A map might be a "hash" consisting of a mapping from textual names to further values (either datum or further type of maps).

So the final structure of data looks like a tree of maps ending in leafs of datum. There are no

balance requirements and in the simplest case the only part of the tree is just the only datum. This means that data can be described in a hierarchical fashion, with values further up in the hierarchy being maps to the ones in lower parts of it.

The supported types of single datum are:

Real numbers

A single number with possible fractional components. These can also be undefined, the meaning of which varies by context.

Integers

A single integer. These can also be undefined, the meaning of which varies by context.

Strings

A string of text. In some contexts this can also have locale specific values (meaning it may appear differently depending on the system language setting). Strings can contain any unicode character, but some contexts are not able to display all of them (e.g. a text only console).

Boolean

A value that can be either true or false.

Flags

Zero or more strings of text usually representing flags. Each string may only be present once in the flags. A string present indicates that the flag is "set".

Binary data

Zero or more bytes of unprocessed data.

Invalid

A special type of value indicating that the value is explicitly undefined and typeless. The most common occurrence of invalid values is when accessing those that are not defined at all in the hierarchy. Invalid values can also be used to undefined the entire hierarchy below them, when used as part of the transparency stack.

The second type of values are those that map from a one type to zero or more further values. The supported type of mapping values are:

Arrays

A list of contiguously indexed values. That is, an array consists of zero or more values in an ordered sequence and the map is from the position in the list to the value. In most contexts arrays are considered to be index from zero: the first element in the list is zero, the second is one, etc.

Matrix

Similar to an array but with two or more dimensions. For example, a two dimensional matrix can be thought of as an array of arrays (of equal size). Matrices do not support ragged elements. That is, all rows, columns, or further dimensions are the same size.

Hash

This is the name used by the system to represent a mapping from a string of text to values. This can also be called an associative array. A hash can also contain a default value (an empty string) that is used when a lookup is performed and does not match any explicitly defined ones. Each string of text maps to exactly one value (no duplicate keys allowed).

Keyframe

This is the name used to representing a mapping from a (defined) real number to values. Keyframes are used when the order of the numbers matters. Each real number maps to exactly one value (no duplicate keys allowed).

There is also a special type of value called an "overlay". An overlay is similar to a datum (in that it cannot itself have values under it), but instead of representing a final value, it instructs the system to insert an entire hierarchic where it is as a replacement. The overlay itself is a path name (described below) that causes the existing contents at that path to replace the overlay before evaluation. This allows for parts of the hierarchy to be reused without duplicating them in definitions.

Finally there are special types of values to represent metadata about other values. These are used primarily in the metadata archive to describe normal values. However, they can also appear in normal configuration that creates other data. For example, the acquisition configuration describing the raw values might include metadata to be written out with them. For all types of metadata there is a mapping from text to metadata values in the same way a hash maps text to values. Hash and keyframe metadata may also include value specific mappings. For example, hash metadata may have metadata values about specific text in the data hashes.

Paths and Values

In order to describe all this in a concise fashion the hierarchy of values can be described a way similar to file paths on an operating system. In this analogy the final datum values are the "files" while the series of maps leading up to them are the "directories". Continuing this analogy, a syntax similar to Unix style path names can be used. This means separating the path components with / and escaping forbidden characters (with \) as required. Using this, a single datum is described by a pair of the path (its "location") and the actual value. The syntax the system common uses has this pair separated by a comma.

Example 1. Simple path definition

```
/Profiles/aerosol/DisableDefaultComponents,TRUE
```

Datum Value Types

The simple datum types above can all be defined using strings of characters. As mentioned above, this is most commonly used in the flat text configuration syntax.

Real Numbers

Values that are real numbers are simply interpreted from a normal text representation of the number.

Example 2. Simple real number

```
2.3
```

Values support both traditional definitions as well as scientific notation (using "e" notation).

Example 3. Scientific notation

```
2.5E-3
```

In order to disambiguate real numbers from integers, the fractional part can always be used (even if it is zero).

Finally real numbers can also be set to undefined by setting them to the special value `NaN` or a metadata defined missing value code, if available.

Integers

Integer values are simply a single integer number interpreted from a text representation.

Example 4. Simple integer

```
42
```

Additionally values may be prefixed with (case insensitive) strings to indicate the base which is used to interpret the number.

`0x`

Base 16 (hexadecimal).

`0o`

Base 8 (octal).

`0i`

Base 10 (decimal).

`0b`

Base 2 (binary).

Example 5. Hexadecimal integer

```
0xAB12
```

In order to disambiguate integer from real numbers, the prefix `0i` can be used to explicitly specify

base ten.

Finally integers can also be set to undefined by setting them to the special value `iNaN` or a metadata defined missing value code, if available.

Text Strings

In the simplest form a text string is just zero or more unicode (UTF-8, usually) characters enclosed in double quote characters (`"`).

Example 6. Simple string

```
"A string of text"
```

In order to allow strings that have quotes or other special characters (for example newlines), the input allows for `\` C-like escaping of characters. For example `\"` produces a double quote character without ending the string. Similarly `\n` translates to a newline character.

Example 7. String with double quotes

```
"A string of \"quoted\" text"
```

A string can also have locale specific values that are substituted base on the system language setting. These are specified by prefixing the locale specific values with the locale name, after the initial default locale-less value.

Example 8. Locacle specific string

```
"The default value" en_US"US english only value"
```

Boolean

Boolean values can be either true or false. Values are case-insensitive.

TRUE, T, ON, or YES

Set the value to true.

FALSE, F, OFF, or NO

Set the value to false.

A non-zero integer

If this is not ambiguous with an integer, then set the value to true.

A zero valued integer

If this is not ambiguous with an integer, then set the value to false.

Example 9. True value

```
TRUE
```

Example 10. False value

```
FALSE
```

Flags

Flags consist of zero or more unique strings that are combined to generate a list of "set" flags. Each individual flag is separated from the others by a `|` character. The entire list of flags also begins with another `|` character.

Example 11. Set of flags

```
|Flag1|Flag2
```

Example 12. Empty flags

```
|
```

Binary Data

Binary data are represented with base-64 encoded values enclosed in a set of `{` and `}`.

Example 13. Binary data value

```
{QmluYXJ5IGRhdGE=}
```

Invalid Values

Explicitly invalid values can be designated with a single `_` (an underscore).

Example 14. Explicit invalid value

```
_
```

Overlays

Overlays are described by a path (see below) prefixed with a `~` (a tilda character). Everything after the tilda is the path that is overlaid in place of the value.

Example 15. Overlay specification

```
~/Templates/Profiles/Default
```

Explicit Type

A final form of datum is an operation that simply sets the type of the value. This will set normal datum values to their empty or undefined state if they did not already exist or where a different type. The more common case is using it to set the type of a mapping value that has nothing under it. This is required, since such empty values would not otherwise appear in a flat representation of the hierarchy.

Explicit types are set by a value starting with = (an equals) followed by a case-insensitive name of the type. The types recognized are:

`real, number, double, or d`

A real number.

`integer or i`

An integer.

`boolean, bool, or b`

A boolean value.

`text, string, or s`

A boolean value.

`binary or n`

Binary data.

`array or a`

An array.

`matrix or t`

A matrix.

`keyframe or t`

A keyframe.

`hash or h`

A hash map.

`flags or f`

Flags.

`*r, mr, metanumber, or metareal`

Real number metadata.

`*i`, `mi`, `metainteger`, **or** `metaint`

Integer metadata.

`*b`, `mb`, `metaboolian`, **or** `metabool`

Boolean metadata.

`*s`, `ms`, `metastring`, `metatext`

Text string metadata.

`*n`, `mn`, **or** `metabytes`

Raw bytes data metadata.

`*a`, `ma` **or** `metaarray`

Array metadata.

`*e`, `me`, **or** `metakeyframe`

Keyframe metadata.

`*h`, `mh`, `*c` **or** `metahash`

Hash map metadata.

`*f`, `mf`, `*l` **or** `metaflags`

Flags metadata.

`invalid`, `undefined`, **or** `undef`

An explicitly undefined value. The same as a simple `_` as described above.

Example 16. An empty hash value

```
=HASH
```

Note that in most cases the above accept the same prefix type values as the path syntax described below.

Hierarchy Paths

In order to designate a particular datum within the hierarchy a filesystem-like path is used as described above. The path itself consists of components separated by `/` characters. Each component determines the specific leaf within that element of the path. A special "component" of `..` (two periods) indicates to move the path up one level. That is, the path will walk up one level when a simple `..` is encountered instead of going down one level. Components of the path obey `\` escaping. For example, `*` is interpreted as just `*`. Path of just an escaped underscore (`_`) is interpreted as the default (empty) hash key. This can be used to allow paths that start with reserved prefixes (see below).

In general the path begins with a `/` indicate the very top of the hierarchy. It is possible to use relative paths (i.e. ones that do not start with a `/`) in which case the path is evaluated from the parent in question. This type of syntax generally only makes sense in overlays beginning with `...`

General Path

If the component of the path does not match any of the other definitions it is interpreted as a hash map key. This is the most common case and allows most paths to just be simple specifications of keys separated by / characters.

Example 17. Hash path

```
/A/Path/With/Hashes
```

Array

Array path components are integers prefixed with # designating an index (from zero) into an array.

Example 18. Array path

```
/#0/#1/#2
```

Matrix

If the path consists of one or more integers separated by :, :, or , (with escaping if required) and surrounded by [and] then it is a matrix path, which the indices in ascending order each starting from zero.

Example 19. Matrix path

```
/[0:1]/[0:1:2]
```

Keyframe

If the path is a real number prefixed with @ then it is a point in a keyframe map.

Example 20. Keyframe path

```
/@0.1/@0/@-3.5
```

Metadata

Metadata paths are similar to general paths, except that they contain a prefix starting with * and a single character indicating the metadata type.

The supported prefix types are:

*r

Real number metadata.

*i

Integer metadata.

***b**

Boolean metadata.

***s**

String metadata.

***n**

Binary value metadata.

***a**

Array metadata.

***t**

Matrix metadata.

***e**

Keyframe metadata.

***h**

Hash map metadata.

***f**

Flags metadata.

***c**

Specific hash value metadata.

***l**

Specific flag metadata.

Example 21. Metadata path

```
/*hHashMetadata/*cHashValueMetadata/*rFormat
```

Additionally a single `^` can be used to specify "any" type of metadata. This is not normally used in the internal syntax, but can be used for user specification when the incoming type of metadata is not known ahead of time.

Example 22. Generic metadata

```
/SomePrefix/^AnyMetadataType
```

Constructed Value Types

Enumeration

An enumeration value is a [text string](#) chosen from a set of possible values. This is used to represent a choice of exactly one of multiple possible options. For example, the units of logical time is a type of enumeration. Enumeration values are not case sensitive.

Since this is just a text string with a restricted set of recognized values, it is represented as a simple value.

Enumeration Value

```
"Minute"
```

Time Interval

A time interval specification is a hash type that is used to specify some block or offset in time based on logical units. It consists of a unit, count and alignment flag. However, in some contexts the alignment or count may be ignored. For example, a time interval may be "one minute", meaning that it applies a change in time by adding a single minute to the input.

The structure of the time interval itself is a hash map type with three possible keys under it.

Units

The units the time interval is specified in. This is an [enumeration](#) type with the possible values:

- Millisecond
- Second
- Minute
- Hour
- Day
- Week
- Month
- Quarter
- Year

Count

The [integer](#) number of units to apply.

Align

The [boolean](#) alignment flag. When set to true, the resulting time of applying the time interval is aligned to a unit boundary.

An additional short hand form of time intervals is available by replacing the entire key with a single integer. In this form the number specifies the number of seconds with the alignment being implied.

One minute time

```
/Units, "Minute"  
/Count, 1
```


Three days aligned

```
/Units,"Day"  
/Count,3  
/Align,TRUE
```

Short hand five minutes

```
/,300
```

Calibration Polynomial

A calibration polynomial is a type of array value whose keys represent the [real number](#) coefficients of a polynomial. This polynomial is generally used to convert a raw signal to physical units or to apply a correction to another value. The coefficients are listed in ascending power order, starting from the offset (x^0). So the coefficient at [#0](#) is the intercept, the one at [#1](#) is the slope, the one at [#2](#) is x^2 , etc.

Intercept 0.25 and slope 0.9

```
/#0,0.25  
/#1,0.9
```

There are no limits to the number of coefficients involved in the calibration, but in most cases only a slope and intercept are utilized.

High order calibration

```
/#0,0.25  
/#1,0.9  
/#2,0.05  
/#3,-0.04
```

Data Sequence Selection

A sequence selection is a specification of the [sequence name](#) component of the data. That is, a sequence selection only selects the name of the sequence with the times depending on the referencing context. In most contexts there are implied defaults for the matched components. For example, a station and archive is often implied by the context the sequence selection is used in.

In the most general form, a sequence selection is an array of hash values. Each element of the array specifies a single selection and all elements are combined together to create the final selection. For example, one element to select each of several variables.

In this context each element of the array is a hash with key values specifying the sequence name components.

Station

The `text` station code regular expression. This regular expression is matched against the entire station, ignoring case. So, if it contains no special characters, it can be treated as the full name of the station.

Archive

The `text` archive regular expression. This regular expression is matched against the entire archive name, ignoring case. So, if it contains no special characters, it can be treated as the full name of the archive.

Variable

The `text` variable regular expression. This regular expression is matched against the entire variable name and is case sensitive. So, if it contains no special characters, it can be treated as the full name of the variable.

HasFlavors

An array of `text` required flavors regular expressions. This is an array of regular expressions that are matched against the present flavors in the sequence name. All elements of the array are required to match at least one present flavor. This can also be simplified into simply a string (instead of an array), which is split by `:` then converted as if it was an array.

LacksFlavors

An array of `text` excluded flavors regular expressions. This is an array of regular expressions that are matched against the present flavors in the sequence name. If any element of the array matches a present data flavor, then the sequence name does not match the selection. In most contexts, this has a default of excluding the "cover", "stats" and "end" flavors, unless override. This can also be simplified into simply a string (instead of an array), which is split by `:` then converted as if it was an array.

Flavors

An array of `text` exact flavors regular expressions. This is an array of regular expressions that are matched against the present flavors in the sequence name. The required flavors must match the sequence name flavors on a one-to-one basis. This can also be simplified into simply a string (instead of an array), which is split by `:` then converted as if it was an array.

Full selection

```
/#0/Station,"bnd"  
/#0/Archive,"raw"  
/#0/Variable,"T_S11"  
/#0/HasFlavors/#0,"pm1"
```

Multiple regular expression

```
/#0/Variable,"(T|P)_S11"  
/#1/Variable,"Q_A11"
```

The first alternate form of the sequence selection omits the array. In this form, it is treated as a single matching element with the above keys valid.

Simple component match

```
/Variable,"BsG_S11"  
/Archive,"clean"
```

Another alternate form is simply a single [text](#) element specifying a combined selection. This is treated much like the command line selection syntax. In the most basic and most common form, this is simply a regular expression matching the variable component of the sequence name. However, it can be elaborated into the various components by splitting it with `:`. In this context, if it has two components, the first is the archive and the second the variable. If it has three, the first is the station, the second the archive and the third is the variable. If it has four or more components, the first two are the same and any subsequent ones are flavor selections. If a flavor selection has no leading character or begins with `+` then it is added to the required flavors. A flavor starting with `-` is added to the excluded flavors and one with `=` is added to the exact flavors, which then take precedence.

Basic text

```
"T_S11"
```

Complex text

```
"bnd::(T|P)_S11:pm1"
```

Dynamic Real Value Input

A dynamic single value input is a means of specifying a single real valued input to a calculation. This consists of a selection from the data, a calibration applied to it and a fallback constant value. In that configuration, the data value is read first, the calibration applied and if the result is defined, that is the final value. If the result is not defined, then the fallback constant value is returned. An alternative configuration is simply a constant value that ignores all input data.

The structure of the time interval itself is a hash map type with keys specifying the above components.

Input

The [variable input](#) from the data stream, before calibration. In most contexts, the station and archive are implied, so only a variable needs to be specified.

Path

The [text](#) specifying the path within the variable to look up. For example, this can be used to select a specific index with an array by setting it to `#3`.

Calibration

The [calibration](#) applied to the data input.

Default

The [real number](#) result returned if no valid value is available from the data after path and

calibration application. This is used to provide a fallback value for when the dynamic values in the data are unavailable.

Constant

If this **real number** is defined, then it is always the result. That is, when this value is always the result and all inputs from data are ignored.

An alternative form of the input sometimes used is simply a single **real number**. This is equivalent to setting the **Constant** option above. That is, when the input is a single real number, then that real number is always returned.

Basic data input

```
/Input/Variable,"T_S11"
```

Calibrated data input with fallback

```
/Input/Variable,"Q_A11"  
/Calibration/#0,0.1  
/Calibration/#1,1.0  
/Default,0.0
```

Final value from a complex type

```
/Input/Variable,"ZSPOT_A11"  
/Path,"/Area"
```

Constant value

```
/Constant,1.0
```

Constant alternate form

```
23.0
```

Baseline Smoother

A baseline smoother is a construct that is used to generate a smoothed value based on prior input values. Then, depending on context, this smoothed value is used as an output and/or as part of spike or stability detection. That is, the smoother generates a value looking backwards in time as well as indicators about the state of the latest value (if it's a spike or the the smoother is now stable).

The basic structure of the smoother consists of a type selection and various parameters for certain types. The parameters for any type not currently selected are not valid and are ignored. In the cases where the smoother is not used for spike and/or stability detection, the associated parameters are also ignored.

The type of the smoother is defined by the **enumeration** value **Type**:

FixedTime

Use a fixed maximum number of seconds worth in a "boxcar" style average. That is, once the smoother reaches its maximum capacity, older points are shifted out in the order they were received. This type of smoother is considered unstable until it receives at least a certain minimum number of seconds worth of data (defaulting to the maximum time). After that minimum, stability is detected by if the total relative standard deviation (σ/μ) is less than a threshold. Spikes are detected by comparing the incoming value to a band determined by the prior smoothed value and if the point is outside the band, it is considered a spike.

For fixed time smoothers, the following parameters are valid:

Time

The **real number** maximum time in seconds the smoother considers.

MinimumTime

The **real number** minimum time before the smoother is considered valid. When set to undefined, the total time is used; the boxcar must be full.

RSD

The **real number** relative standard deviation threshold used to determine if the smoother is stable or not. When set to undefined, the smoother is always stable.

Band

The **real number** relative band of the smoothed value used for spike detection. For example, a value of 0.5 means that any value outside the range of half to twice the smoothed value is considered a spike.

SinglePole and FourPole

These smoother types are low pass digital filters of one and four poles, respectively. A low pass digital filter is an exponential infinite response type of filter. That is, they consider the prior smoothed value(s) weighted against the latest incoming ones to generate an output smoothed value. Both forms of filter accept the same parameters and are defined in terms of a time constant. This time constant is the same as an RC time constant. Conceptually, this represents the time it would take for the smoothed output value to reach 63.2 percent of the final value after a step change in the input.

Digital filter type smoothers detect spikes and stability by comparing the incoming value to relative bands of the prior smoothed one. For example, a stable band of 0.5 means that the smoother is stable as long as the incoming value is within half to twice the smoothed one.

For both forms of digital filter, the following parameters are valid:

TimeConstant

The **time interval** representing the time constant (impulse response time) of the filter.

Gap

The **time interval** representing the maximum time between two incoming points before the smoother is reset. When set to undefined, no reset is performed; the value is considered infinite.

ResetOnUndefined

When the **boolean** value is true, the smoother is reset as soon as an undefined value is received.

StableBand

The **real number** relative band of the smoothed value used for stability detection. The smoother is stable as long as the incoming value is within this band of the prior smoothed one.

Spike

The **real number** relative band of the smoothed value used for spike detection. The incoming value is considered a spike if it is outside of this band of the smoothed value.

SinglePoint

Use the latest valid (defined) single value. Single point smoothers are stable if they have received a valid value. They do not perform spike detection.

Latest

Use the latest value, regardless of if it's defined or not. Latest point smoothers are stable if the latest point is defined. They do not perform spike detection.

Disable

Disable all smoothing and report only invalid values. When disabled, the smoother is never stable, ready or a spike.

Forever

Use all valid points the smoother has received in an arithmetic mean average. The smoother is stable as soon as it as received a valid value. No spike detection is performed.

90 second maximum and 30 second minimum smoother

```
/Type,"FixedTime"  
/Time,90.0  
/MinimumTime,30.0
```

Single value smoother

```
/Type,"SinglePoint"
```

SSL/TLS Definition

A SSL definition sets the parameters need to accept or receive encrypted SSL or TLS connections. Generally this includes a local certificate and private key to use with the connection. It may also include a certificate authority definition to validate the peer's certificate against.

The structure of the specification itself is a hash map type with multiple possible keys below it.

Certificate

The the local certificate. This can either be the full **text** contents of the certificate in PEM format or the **binary** contents in DER format. In contexts that permit external loading, it may also be the

`text` of a file name to load containing either PEM or DER data.

Key

The local private key. This can either be the full `text` contents of the key in PEM format or the `binary` contents in DER format. In contexts that permit external loading, it may also be the `text` of a file name to load containing either PEM or DER data.

Authority/#0

An array of certificates that compose the certificate authority to validate against. Each certificate can either be the full `text` contents of the certificate in PEM format or the `binary` contents in DER format. In contexts that permit external loading, it may also be the `text` of a file name to load containing either PEM or DER data.

NoLocal

A `boolean` setting that explicitly disables the local certificate and key.

RequireValid

A `boolean` setting that enables validation against the certificate authority of the peer's certificate. When set, the peer must possess a certificate that verifies against the authority or the connection is rejected.

MinimumVersion

A `integer` specifying the minimum TLS version accepted. For example, setting this to two requires TLS 1.2 or higher. Setting it to -1 accepts any TLS version.

Acquisition I/O Interface

An acquisition I/O interface is a specification of an input and output target. This target is generally an external piece of hardware that the system is communicating with. The most common example is a specification of a serial port to which an instrument is attached.

The basic structure of the interface consists of a type selection and various parameters for certain types. The parameters for any type not currently selected are not valid and are ignored.

In addition to input and output streams of data, I/O interfaces have a concept of "echo". That is, they can also report output written by other uses of the same interface. This is normally used by instruments operated in read-only mode while listening to another source controlling the actual hardware. For example, it is possible to have a CPD3 Love PID controller listening to CPD2 controlling the actual hardware and while using the echo stream it can interpret the command responses.

In some contexts, the whole specification of the interface is not required. For example, the serial port parameters can be omitted for most acquisition contexts, because they are defined by the type of instrument actually connected. Similarly, blacklisting or adding ports for auto-detection does not require the serial port parameters, since they only match the hardware port definition.

The type of the interface is defined by the `enumeration` value `Type`:

SerialPort

A locally connected physical serial port. This is an input and output interface defined by the OS

level name of the serial port.

Port

The [text](#) OS level name of the serial port. For example `"/dev/ttyUSB0"` on Linux or `"COM1"` on Windows.

Baud

The [integer](#) baud rate of the serial port.

Parity

The [enumeration](#) parity bit behaviour selection. One of `"None"`, `"Even"` or `"Odd"`.

DataBits

The [integer](#) number of data bits in a single frame.

StopBits

The [integer](#) number of stop bits in a single frame.

HardwareFlowControl

When this [boolean](#) is enabled, the serial port will use the RTS and CTS signal lines for flow control.

SoftwareFlowControl

The [enumeration](#) software flow control setting. One of `"None"` (XON/XOFF are normal characters; the default), `"Bidirectional"` (XON/XOFF used in both input and output), `"Input"` (input stream only), or `"Output"` (output stream only).

SoftwareFlowControlResume

When this [boolean](#) is enabled, flow control will resume on reception of any character (not just XON). Meaningless unless input software flow control is in use.

NoInitialBreak

When this [boolean](#) is enabled, the initial break (line held in mark condition) on opening the serial port is omitted.

Eavesdropper

The [text](#) OS level name of the serial port to read eavesdropper echo data from.

RemoteServer

A TCP connection to a remote server. This type of interface passes input and output data over a TCP connection initiated to a provided server and port. The most common use of this is to connect to a "serial to ethernet" type of converter box. However, it can also be used to connect to any type of general TCP server to exchange data.

Server

The [text](#) IP address or host name to connect to.

ServerPort

The [integer](#) port number to connect to.

SSL

The [ssl specification](#) used with the connection.

CPD3Server

A TCP connection to a CPD3 style I/O interface server. This functionality allows an instance of the I/O server to be started on a remote machine then another one to connect to it and control a local interface. That is, this can be used as a replacement for a USB to IP converter type box, using a full computer to do the conversion. For example, the machine with the hardware connected to it would start an I/O server on the connected ports, while having its configuration specify a valid remote server. Then another one would have a `CPD3Server` style interface pointed at the first one. This interface then is allowed to change serial port parameters and reconfigure the local hardware as required.

Server

The `text` IP address or host name to connect to.

ServerPort

The `integer` port number to connect to.

SSL

The `ssl specification` used with the connection.

Command

Run an external command and communicate with it via its standard input and output channels. This allows for an external command to be a sink and/or source of data to an I/O interface. Note that the command may be restarted outside of its control (whenever the I/O interface re-initializes). It should also not use buffered output or should take care to flush the buffers or the I/O interface will not receive data in a timely fashion.

Command

The `text` name of the executable to invoke.

Arguments/#0

An array of `text` arguments passed to the command. For example, `Arguments/#0` is the first argument and `Arguments/#1` is the second.

TCPListen

A server that listens for incoming TCP connections then exchanges data over any that are open. This type of interface waits for an incoming TCP connection then will and and receive data to any connections that are currently open.

LocalPort

The `integer` port number to listen on.

ListenAddress

The `text` IP address of the network to restrict listening on. If set the the socket will only accept connections for the network interface associated with the given IP address.

SSL

The `ssl specification` used with the connection.

SSL/Authorization/#0

An array of SSL certificates permitted to connect. When not present, only the basic SSL

constraints are enforced.

UDP

This interface sends and receives data via UDP datagram packets. This type of interface accepts UDP packets for incoming data and takes their contents as a single input stream. Likewise, it packages outgoing data into packets and sends them to a remote host. Note that since the internal view of data is stream based, not packet based, the contents of where the UDP packets are split may not always coincide with single send and receives.

Server

The [text](#) IP address or host name to send data to.

ServerPort

The [integer](#) destination port number to send data to.

LocalPort

The [integer](#) local port to listen for incoming packets on.

ListenAddress

The [text](#) IP address of the network to restrict listening on. If set the the socket will only accept connections for the network interface associated with the given IP address.

FragmentSize

The [integer](#) maximum size of a single packet before it is split.

Multiplexer

A multiplexer interface combines multiple sub-interfaces into a single composite one. It allows for arbitrary mixing of input and output channels into a final view. For example, it can be used to combine the input from two serial ports into a single logical stream. Similarly, it can be used to construct an interface that sends data out one serial port but receives data from another physically distinct one.

Since the multiplexer only forwards data and not parameters, all sub-interfaces must provide all parameters needed to configure themselves. For example, serial ports should provide a baud rate.

Targets/Input

The [set](#) of group names that generate input for the composite interface.

Targets/Output

The [set](#) of group names that outgoing data is assigned to.

Targets/Echo

The [set](#) of group names that generate echo data for the composite interface.

Interfaces/#0

An array of interface definitions that the multiplexer composites.

Interfaces/#0/Channels/Input

The [set](#) of groups that input data on the sub-interface is assigned to.

Interfaces/#0/Channels/Output

The **set** of groups that are matched against the composite ones when generating output.

Interfaces/#0/Channels/Echo

The **set** of groups that echo data on the sub-interface is assigned to.

QtLocalSocket

A bidirectional connection to a machine-local Qt-managed socket.

Name

The **text** name of the socket to connect to.

QtLocalListen

A bidirectional connection from a machine-local Qt-managed socket. This creates a socket that listens for other machine local communications then sends and receives data over all open connections.

Name

The **text** name of the socket to connect to.

LocalSocket

Only supported on Unix. A bidirectional connection to a local Unix domain socket.

File

The **file name** of the Unix domain socket.

LocalListen

Only supported on Unix. A bidirectional connection from a local Unix domain socket. This creates a socket that listens for other machine local communications then sends and receives data over all open connections.

File

The **file name** of the Unix domain socket.

Pipe

Only supported on Unix. This type of interface communicates with a set of Unix named pipes (FIFOs) that represent the three channels of communications. That is, it send data to the output pipe, and reads input and echo from their respective pipes.

Input

The **file name** of the Unix named pipe to read input data from.

Output

The **file name** of the Unix named pipe to write output data to.

Echo

The **file name** of the Unix named pipe to read echo data from.

Serial port with automatic parameters

```
/Type,"SerialPort"  
/Port,"/dev/ttyUSB1"
```

Serial port with manual parameters

```
/Type,"SerialPort"  
/Port,"/dev/ttyUSB1"  
/Baud,115200  
/Parity,"None"  
/StopBits,2  
/DataBits,8
```

Remote server connection

```
/Type,"RemoteServer"  
/Server,"192.168.0.111"  
/ServerPort,12341
```

Local command communication

```
/Type,"Command"  
/Command,"cat"  
/Arguments/#0,"-"
```

Split multiplexer output

```
/Type,"Multiplexer"  
/Targets/Input,|in  
/Targets/Output,|out  
/Interfaces/#0/Channels/Input,|in  
/Interfaces/#0/Type,"SerialPort"  
/Interfaces/#0/Port,"/dev/ttyUSB0"  
/Interfaces/#0/Baud,9600  
/Interfaces/#1/Channels/Output,|out  
/Interfaces/#1/Type,"SerialPort"  
/Interfaces/#1/Port,"/dev/ttyUSB1"  
/Interfaces/#1/Baud,9600
```

Color Definition

A color defines both the physical color of an element on the screen or in a plot as well as the transparency of that element. For example, a red color with full opacity will always draw a red element. However, one with half transparency will blend the color "below" it with red using half of each. The transparency is selected by the **real number A** which ranges from zero (fully transparent) to one (fully opaque). If the **A** key is not specified, it is assumed to be one for full opacity.

Color definitions also include the color model used to define the actual color. By default, this is the standard *RGB* (red, green, and blue) channels used with most computer graphics. While the alpha or transparency is common to all color models, the specific channels used depend on the one selected.

The color model in use is selected by the [enumeration](#) value `Model`:

RGB

The default color model consisting of channels for the red, green, and blue components of the color.

R

The [real number](#) of the red channel. This ranges from zero (none) to one (maximum).

G

The [real number](#) of the green channel. This ranges from zero (none) to one (maximum).

B

The [real number](#) of the blue channel. This ranges from zero (none) to one (maximum).

HSV

A color model that specifies the hue, saturation, and value of the color.

H

The [real number](#) of the color hue. This ranges from zero to one, specifying the spectrum of the color.

S

The [real number](#) of the color saturation. This ranges from zero to one, specifying how close to white the hue is.

V

The [real number](#) of the color value. This ranges from zero to one, determining how bright the color is.

HSL

A color model that specifies the hue, saturation, and lightness of the color.

H

The [real number](#) of the color hue. This ranges from zero to one, specifying the spectrum of the color.

S

The [real number](#) of the color saturation. This ranges from zero to one, specifying how close to white the hue is.

L

The [real number](#) of the color value. This ranges from zero to one, determining the color whiteness. Pure colors have a lightness of 0.5, with white at one and black at zero.

CMYK

A color model that specifies the cyan, magenta, yellow, and black components of the color. This

color model corresponds to various print color specifications.

C

The **real number** of the cyan channel. This ranges from zero (none) to one (maximum).

M

The **real number** of the magenta channel. This ranges from zero (none) to one (maximum).

Y

The **real number** of the yellow channel. This ranges from zero (none) to one (maximum).

K

The **real number** of the black channel. This ranges from zero (none) to one (maximum).

Red

```
/R,1.0
```

Yellow with 25% transparency.

```
/Model,"RGB"  
/R,1.0  
/G,1.0  
/A,0.25
```

Green in HSV

```
/Model,"HSV"  
/H,0.47  
/S,1.0  
/V,1.0
```

Gray and pale red in HSL

```
/Model,"HSL"  
/H,0.0  
/S,0.2  
/L,0.8
```

Dark blue in CMYK

```
/Model,"CMYK"  
/C,1.0  
/M,1.0  
/Y,0.0  
/K,0.5
```

Color Gradient

A gradient is a means of mapping real numbers to colors. For example, the color in a time series trace of binned size distribution is determined by a gradient applied to the counts.

There are several standard gradients available as well as the capability to define a custom one. Custom gradients are controlled by defining keyframe colors at specific points in the gradient. The output color is then the interpolation of the two bracketing color definitions.

The gradient type is selected with the `enumeration` value `Type`:

Sequential **or** Default

The default standard gradient suitable for sequential data. This is best for data that covers the whole range with equal importance. It also provides visibility even with reduced color perception.

Diverging **or** Diverge

A gradient suitable for data that diverges. This gradient exaggerates the difference between the limits of the data, with reduced difference near the middle of the range. It also provides visibility even with reduced color perception.

Rainbow

A gradient that spans a wide spectrum of colors. This is suitable for making very colorful plots, with some reduction in visibility with reduced color perception.

AlternateRainbow **or** AltRainbow

An alternate form of a spectrum spanning gradient, with the same limitations

Manual

A manual gradient definition. This form of gradient requires specific points in the normalized [0,1] space to be assigned to colors. The output color is then the interpolation between the two color points that bracket the original input value.

Colors

A keyframe mapping the normalized value [0,1] to the output `color`.

Mode

An `enumeration` specifying the color model the interpolation is performed in.

RGB

Interpolate the red, green, and blue components.

HSV

Interpolate the hue, saturation, and value components.

HSL

Interpolate the hue, saturation, and lightness components.

CMYK

Interpolate the cyan, magenta, yellow, and black components.

Model

An [interpolation model](#) defining how the interpolation is performed, if not purely linear.

Basic sequential

```
/Type, "Sequential"
```



Diverging data

```
/Type, "Diverging"
```



Rainbow gradient

```
/Type, "Rainbow"
```



Alternate rainbow gradient

```
/Type, "AlternateRainbow"
```



Manual gradient from red to green then to blue

```
/Type, "Manual"  
/Colors/@0.0/R,1.0  
/Colors/@0.5/G,1.0  
/Colors/@1.0/B,1.0
```



Interpolation Model Definition

An interpolation model is a means of converting a set of input points into some general function. For example, a linear regression is one type of model that does not necessarily evaluate to all points exactly. Another type of model is a cubic spline interpolator, which does evaluate to all points exactly, but provides smoothed curves between them. Through all model types are available in all contexts, they may not necessarily make sense in some.

Interpolation models provide multi-dimensional interpolation, but many of the specific types simply operate with each dimension independently. For example, a model used in a gradient definition may simply use linear interpolation of each component of the color independently. In most cases, the first dimension is used as the independent variable.

The model in use is selected by the [enumeration](#) value `Type`:

`LeastSquares` or `LSQ`

A least squares linear regression of the input points.

`ZeroLeastSquares` or `ZLSQ`

A least squares linear regression of the input points, with the intercept forced through zero.

`NPolynomial` or `NPoly`

An N-th order polynomial, fitted with non-linear least squares (Levenberg-Marquardt).

`Order`

The [integer](#) order of the polynomial. For example, two gives a polynomial of the form $a + b*x + c*x^2$

`ZeroNPolynomial` or `ZNPoly`

An N-th order polynomial with no intercept term, fitted with non-linear least squares (Levenberg-Marquardt).

`Order`

The [integer](#) order of the polynomial. For example, two gives a polynomial of the form $a*x + b*x^2$

`Linear`

A linear interpolator between all input data points. This creates a piecewise function that starts with a linear interpolation between each point and then (optionally) applies a transfer function.

`Transfer`

An [enumeration](#) specifying the transfer function applies to each segment.

`Linear`

Use the default of no transfer function. The output is simply the input line segment.

`Sin`

A sin based function of the form $(\sin(x))^n$. This creates a "concave" form of interpolation, with the greatest rate of change near the start.

Cos

A cos based function of the form $(1-\cos(x))^n$. This creates a "convex" form of interpolation, with the greatest rate of change near the end.

Sigmoid

A sigmoid shaped function. The sigmoid created is symmetric around the middle of the range with the parameter in $[-1,1]$ with negative values causing increased rate of change at the start end end.

TransferParameter

The **real number** of the parameter to the transfer function. This is the power in the sin and cos variants and the tunable input in the sigmoid.

CubicSpline or CSpline

A cubic spline interpolator between all points. This produces a smooth interpolator between each point.

Clamp

The **real number** slope of the interpolator at the endpoints, if neither is overridden. If set to undefined, a natural spline is used; meaning that the slope is set such that the second derivative is zero.

ClampStart

The **real number** slope of the interpolator at the first point. If set to undefined, a natural spline is used; meaning that the slope is set such that the second derivative is zero.

ClampEnd

The **real number** slope of the interpolator at the last point. If set to undefined, a natural spline is used; meaning that the slope is set such that the second derivative is zero.

CDF or NormalCDF

A fit line suitable for data that has been transformed into a CDF. The fine line assumes a normal distribution and calculates the mean and standard deviation. If the data are truly normal, the result would be an exactly straight line in the CDF with those parameters.

Allan

A fit line suitable for data that has been transformed into an Allan plot. The fine line is generated between a minimum and maximum averaging intervals and its slope is the reduction in standard deviation from more averaging.

Start

The **real number** of averaging seconds to start the fitting at.

End

The **real number** of averaging seconds to end the fitting at.

Intercept

The **real number** of averaging seconds to report the intercept at in graph legends.

Density

A density plot filling interpolator. This interpolator takes the input points and produces a field

that can be evaluated to generate an image of the relative density of points. The field itself is created by creating gaussian shaped kernels around each point and adding the results together.

Density

The **real number** density parameter. Smaller values give larger spots around each point.

Limit

The **real number** limit factor. From zero to one, this determines the weight limit of a point in the field. Smaller values "clip" peaks in the field.

Constant

A model that always evaluates to a constant value. It can optionally calculate this constant value as the mean of the input values.

Value

The **real number** output. When undefined, the mean of the input values is calculated instead.

GeometricAverage

A **boolean** value that when set causes the mean to be calculated with a geometric average, instead of an arithmetic one.

Composite

This creates a composite model, allowing for the selection of different models for each component or dimension.

DefaultComponent

A model definition that sets the default for the composite model when a dimension is not explicitly defined.

Components

An array of model definitions for the components or dimensions of the input.

Least squares regression

```
/Type, "LeastSquares"
```

Clamped cubic spline interpolation

```
/Type, "CubicSpline"  
/Clamp, 0.0
```

CDF fit line

```
/Type, "CDF"
```

```
/Model,"Composite"  
/DefaultComponent/Type,"Linear"  
/Components/#0/Type,"Linear"  
/Components/#1/Type,"CubicSpline"
```

Data Transformer Definition

A data transformer is a general purpose interface to apply a function to a set of points. The output may alter the number of points by adding or removing some. A simple example of a transformer is one that removes all data outside a range, to limit spikes. Another example of one is a conversion of the input data into a cumulative distribution function.

The transformer type is selected with the [enumeration](#) value `Type`:

Range **or** AcceptRange

Only accept data within a specified range, removing or invaliding all points outside of the range.

Minimum

The [real number](#) minimum accepted value.

Maximum

The [real number](#) maximum accepted value.

MinInclusive

A [boolean](#) that causes the minimum bound to be inclusive. When set, this means that data exactly equal to the minimum are included.

MaxInclusive

A [boolean](#) that causes the maximum bound to be inclusive. When set, this means that data exactly equal to the maximum are included.

Remove

A [boolean](#) that causes the transformer to remove points entirely, instead of simply invaliding them.

Dimension

The [integer](#) dimension, starting from zero, to apply the filter on, instead of all data dimensions.

RejectRange

Reject data within a specified range, removing or invaliding all points inside of the range.

Minimum

The [real number](#) minimum rejected value.

Maximum

The [real number](#) maximum rejected value.

MinInclusive

A **boolean** that causes the minimum bound to be inclusive. When set, this means that data exactly equal to the minimum are included.

MaxInclusive

A **boolean** that causes the maximum bound to be inclusive. When set, this means that data exactly equal to the maximum are included.

Remove

A **boolean** that causes the transformer to remove points entirely, instead of simply invalidating them.

Dimension

The **integer** dimension, starting from zero, to apply the filter on, instead of all data dimensions.

QuantileFilter **or** AcceptQuantile

Only accept data within a specified range of quantiles, removing or invalidating all points outside of the range. The quantiles are the fractions of the the total range. For example, 0.25 specifies the lower 25% quantile of the data.

Minimum

The **real number** minimum accepted quantile in [0,1].

Maximum

The **real number** maximum accepted quantile in [0,1].

MinInclusive

A **boolean** that causes the minimum bound to be inclusive. When set, this means that data exactly equal to the minimum are included.

MaxInclusive

A **boolean** that causes the maximum bound to be inclusive. When set, this means that data exactly equal to the maximum are included.

Remove

A **boolean** that causes the transformer to remove points entirely, instead of simply invalidating them.

Dimension

The **integer** dimension, starting from zero, to apply the filter on, instead of all data dimensions.

RejectQuantile

Reject data within a specified range of quantiles, removing or invalidating all points inside of the range. The quantiles are the fractions of the the total range. For example, 0.25 specifies the lower 25% quantile of the data.

Minimum

The **real number** minimum rejected quantile in [0,1].

Maximum

The **real number** maximum rejected quantile in [0,1].

MinInclusive

A **boolean** that causes the minimum bound to be inclusive. When set, this means that data exactly equal to the minimum are included.

MaxInclusive

A **boolean** that causes the maximum bound to be inclusive. When set, this means that data exactly equal to the maximum are included.

Remove

A **boolean** that causes the transformer to remove points entirely, instead of simply invalidating them.

Dimension

The **integer** dimension, starting from zero, to apply the filter on, instead of all data dimensions.

Wrap **or** RangeWrap **or** WrapRange

A transformer that introduces new points and breaks so simulate "off plot" wrapping of data. This is generally used for data that is circular and should be plotted with the wrapped around points going off the top or bottom of the plot, instead of directly connecting to the next one. For example, wind direction in degrees would wrap from a minimum of zero to a maximum of 360.

Minimum

The **real number** minimum or lower bound of the wrapping range.

Maximum

The **real number** maximum or upper bound of the wrapping range.

CDF **or** NormalCDF

Transform data into a cumulative distribution function. Additionally, the data re normalized into standard deviation units.

Allan **or** AllanPlot

Transform data into points of averaging for an Allan plot. This generated a series of noise estimates from the standard deviation of multiple averages at a series of averaging intervals. The averaging intervals are selected to cover the whole range of available times that still give sufficient variability of a reasonable estimate.

PowerSpectrum **or** Fourier **or** FFT

Perform a Fourier transform to generate a normalized power spectrum. The resulting power spectrum represents the relative strength of the component of the input data when in the frequency (instead of time) domain.

Constant

Set all points to a constant value.

Value

The [real number](#) output value.

NOOP or None

Do not perform any transformation. This is primarily useful as part of a chain or overlay to explicitly disable a transformer.

Chain

Apply a series of transformers to the data. A chain transformer combines multiple transformers by applying them one after another. For example, a filter to remove outliers then a transformation into a CDF.

Chain

The array of transformers to apply, in order.

Simple outlier filtering

```
/Type,"QuantileFilter"  
/Minimum,0.05  
/Maximum,0.95
```

Wind direction wrapping

```
/Type,"Wrap"  
/Minimum,0.0  
/Maximum,360.0
```

CDF data transformation

```
/Type,"CDF"
```

Outlier filtering then CDF generation

```
/Type,"Chain"  
/Chain/#0/Type,"QuantileFilter"  
/Chain/#0/Minimum,0.05  
/Chain/#0/Maximum,0.95  
/Chain/#1/Type,"CDF"
```

Wavelength Adjusted Data Control

A wavelength adjuster is a construct that is used to manage wavelength dependence and adjustment of optical coefficients. The most common case is simply adjusting scattering, absorption, or extinction coefficients from their native wavelengths to some other targets. Another example usage is generating Ångström exponents from pairs of neighboring wavelengths in the data. The adjuster also provides optional smoothing of the inputs before calculation of the output.

When generating output values, the adjuster has the concept of "valid distance". This is the number of nanometers that a given input value can be interpolated or extrapolated to a target wavelength.

So, if the target wavelength exceeds the valid distance from all available inputs, the output is returned as unavailable or invalid. This allows inputs to only provide nearby values if they are approximating to limit the uncertainty of the approximation.

The wavelength adjuster consists of input classes and their parameters. The first class of inputs is the dynamically controlled ones. These inputs are specified by a match of potentially many variables and take their input wavelengths from the metadata of the associated variable. The configuration for dynamic inputs is on the top level of the wavelength adjuster hash configuration.

Input

The **input variables** considered for dynamic input. This is usually a specification that matches all wavelength channels for a given instrument.

ValidDistance

The **real number** valid distance in nm from each inputs.

Smoothing

The **smoothing** applied to each input independently before output generation.

The next set of inputs are single channels at controlled wavelengths. These inputs do not incorporate the available metadata but the actual input value may be overridden. The structures of these inputs are as unique identifiers under the **Fixed** hash in the top level of the adjuster.

Fixed/<IDENTIFIER>/Input

The **input value** of the channel. This should produce a value in Mm^{-1} suitable for adjustment or Ångström exponent calculations.

Fixed/<IDENTIFIER>/Wavelength

The **real number** effective wavelength in nm of the channel. This is the wavelength used for both validity calculations and actual optical calculations.

Fixed/<IDENTIFIER>/ValidDistance

The **real number** valid distance in nm for the input channel.

Fixed/<IDENTIFIER>/Smoothing

The **smoothing** applied to the input channel before output generation.

In addition to the optical coefficient inputs, the adjuster can also have direct Ångström exponent inputs. The first set of these are defined by unique identifiers under the **Angstrom** hash in the top level of the adjuster. These Ångström exponents are preferred for interpolation and explicit calculations if they are provided and within the valid distance. That is, if a value is provided here, the adjuster will attempt to use it before calculating one from any available inputs.

Angstrom/<IDENTIFIER>/Input

The **input value** of the Ångström exponent channel. This should produce the final Ångström exponent, even if it is just a constant.

Angstrom/<IDENTIFIER>/Wavelength

The **real number** central wavelength in nm of the channel. This is used for validity calculations.

Angstrom/<IDENTIFIER>/ValidDistance

The **real number** valid distance in nm for the input channel.

Angstrom/<IDENTIFIER>/Smoothing

The **smoothing** applied to the input channel before output generation.

The final set of inputs are fallback Ångström exponents. These are Ångström exponents used only when the calculation from calculation from the optical coefficients fails. For example, the fall back Ångström exponent can be used when only a single coefficient is available or when channels are negative. The configuration for fallback Ångström exponents are located as unique identifiers under the `AngstromFallback` hash in the top level of the adjuster.

AngstromFallback/<IDENTIFIER>/Input

The **input value** of the Ångström exponent channel. This should produce the final Ångström exponent, even if it is just a constant.

AngstromFallback/<IDENTIFIER>/Wavelength

The **real number** central wavelength in nm of the channel. This is used for validity calculations.

AngstromFallback/<IDENTIFIER>/ValidDistance

The **real number** valid distance in nm for the input channel.

AngstromFallback/<IDENTIFIER>/Smoothing

The **smoothing** applied to the input channel before output generation.

Simple automatic adjustment

```
/Input/Variable,Ba[BGRQ0-9]*_A11
```

Single variable explicit Ångström exponent

```
/Input/Variable,BaG_A11  
/Angstrom/Green/Input,1.0  
/Angstrom/Green/Wavelength,550  
/Angstrom/Green/ValidDistance,NaN
```

Explicit channel definition

```
/Fixed/Green/Input/Variable,BsG_S11  
/Fixed/Green/Wavelength,550  
/Fixed/Red/Input/Variable,BsR_S12  
/Fixed/Red/Wavelength,700
```

Editing Action

An edit action is an operation performed on a stream of data. The simplest case is invalidation: setting all affected data to explicitly missing values.

The action type is selected with the **enumeration** value `Type`:

Invalidate

Invalid data, preserving the presence of it but setting it to explicitly missing values. This causes the data to show up as missing, but without introducing a gap indicating the absence of acquisition.

Selection

The [data selection](#) to invalidate.

PreserveType

A [boolean](#) indicating if the invalidation preserves the type (e.g. real number) or if it sets it to a general undefined indicator. Defaults to enabled so the type is preserved, if possible.

Contaminate **or** Contam

Apply contamination to all data in the stream. This sets a contamination flag indicating that the data are not representative of ambient conditions and should not be averaged into final products.

Origin

The [name](#) of the source of contamination. This is part of the final flag name and defaults to "Mentor".

Description

The [text](#) description of the contamination source. This is recorded in the metadata about the flag and defaults to "Data flagged as contaminated by the station mentor".

Bits

The [integer bits](#) associated with the flag. This is used when generating a combined flags output and defaults to 0x2.

Uncontaminate **or** ClearContam

Remove all contamination flagging in the data stream. This undoes the effect of any prior source of contamination by cleaning all contamination related flags.

FlowCorrection **or** FlowCalibration

Apply a correction to a flow rate and any parameters derived from that flow rate. A flow correction is used when, for example, a calibration on a CLAP was later discovered to be incorrect. To apply a simple revision to an existing flow, use a unity (0, 1) calibration as the original, effectively disabling the normal calibration reversal.

Original

The original, incorrect, [calibration](#) of the flow rate.

Calibration

The final, true, [calibration](#) of the flow rate.

CombineCut

A [boolean](#) indicating if the accumulation handling should span cut sizes, defaulting to enabled.

Minimum

The [real number](#) minimum expected value of the uncalibrated (original reversed) input. This

is used for root selection when reversing higher order polynomials.

Maximum

The **real number** maximum expected value of the uncalibrated (original reversed) input. This is used for root selection when reversing higher order polynomials.

Instrument

The **name** of the instrument to apply to. For example "A11".

Flow

When the instrument is not set, the **data selection** of the flow being corrected.

Selection

When the instrument is not set, the **data selection** of the flow derived values being corrected (e.g. absorptions).

Accumulator

When the instrument is not set, the **data selection** of any accumulated qualities derived from the flow (e.g. sample volume and length).

Spot

When the instrument is not set, the **data selection** of any associated total spot data.

Spot **or** SpotSize

Apply a correction for a filter absorption photometer spot size. This is used to correct for an inaccurately measured spot size on a single spot instrument (e.g. the PSAP-3W). If only a ratio is available, set the original spot size to one and the corrected to the ratio value.

Original

The original, incorrect, **real number** area of the spot in mm².

Corrected

The final, true, **real number** area of the spot in mm².

Instrument

The **name** of the instrument to apply to.

Selection

When the instrument is not set, the **data selection** of the flow derived values being corrected (e.g. absorptions).

Spot

When the instrument is not set, the **data selection** of any associated total spot data.

Length

When the instrument is not set, the **data selection** of any associated accumulated sample length.

MultiSpotSize, MultiSpot, **or** CLAPSpot

Apply a correction for a filter absorption photometer switched spot size. This is used to correct for an inaccurately measured spot size on a switched spot instrument (e.g. the CLAP). If only a

ratio is available, set the original spot size to one and the corrected to the ratio value.

Original

An array of the original, incorrect, **real number** area of the spot in mm².

Corrected

An array of the final, true, **real number** area of the spot in mm².

Instrument

The **name** of the instrument to apply to.

Selection

When the instrument is not set, the **data selection** of the flow derived values being corrected (e.g. absorptions).

Spot

When the instrument is not set, the **data selection** of any associated total spot data.

Length

When the instrument is not set, the **data selection** of any associated accumulated sample length.

Polynomial, Poly, Cal, or Calibration

Apply a simple calibration polynomial to values in the data.

Calibration

The **calibration** to apply.

Selection

The **data selection** to apply the calibration to.

SetCut or Cut

Alter the logged cut size of data. This is used when the acquisition logged cut size does not match the true sampled cut size (e.g. the impactor was stuck). In general, the selection is used to select all data of the faulty cut size. If changing to a cut size that was not possibly active (i.e. in a non-switched system), the metadata application also needs to be enabled.

Selection

The **data selection** to alter.

Cut

The **cut size** to change to. For example, "PM10".

ApplyToMetadata

A **boolean** controlling if the size change also applies to metadata. This is not required if metadata for the target cut size already exists (e.g. a switched impactor system).

Serial or SetSerial

Set the serial number recorded in the metadata. This is used when the data recorded by the system has an inaccurate or incomplete serial number. The effect is purely an alteration of the metadata, no data values are changed.

SerialNumber

The usually [integer](#) serial number. Any data type is accepted here and will be directly copied to the metadata.

Instrument

The [name](#) of the instrument to apply to.

Selection

When the instrument is not set, the [data selection](#) to alter the serial number for.

AddFlag, Flag, or AddFlags

Add one or more flags to existing flags data values. This sets the specified flags in any selected data.

Flags

The [flags](#) to add.

Selection

The [data selection](#) of flags variables to add into.

Description

The [text](#) description of the added flag recorded into the metadata.

Bits

The [integer bits](#) associated with the flag.

RemoveFlag or RemoveFlags

Remove one or more flags from the data stream. This clears the specified flags in any selected data.

Flags

The [flags](#) to remove.

Selection

The [data selection](#) of flags variables to remove from.

RemoveMatchingFlags, RemoveAnyFlag, or RemoveAnyFlags

Remove any flags matching a regular expression pattern from the data stream. This will remove flags based on a pattern match, allowing for wildcard removal of multiple flags without explicitly specifying each one.

Flags

Any array of [text](#) patterns to match.

Selection

The [data selection](#) of flags variables to remove from.

CheckAll

A [boolean](#) that causes each individual flag to instance to be checked, instead of only removing the matching flags defined in the metadata. This is not normally required and is only used if the metadata is incomplete.

PolynomialInvert, PolyInvert, or InvertCal

Reverse or invert a polynomial, transforming the values to what they would be before the polynomial was applied.

Calibration

The **calibration** to invert.

Selection

The **data selection** to apply to.

Minimum

The **real number** minimum expected value of the uncalibrated input. This is used for root selection when reversing higher order polynomials.

Maximum

The **real number** maximum expected value of the uncalibrated input. This is used for root selection when reversing higher order polynomials.

Recalibrate

First reverse a polynomial then apply a new one.

Original

The original, incorrect, **calibration** to reverse.

Calibration

The final, true, **calibration** to apply.

Selection

The **data selection** to apply to.

Minimum

The **real number** minimum expected value of the uncalibrated (original reversed) input. This is used for root selection when reversing higher order polynomials.

Maximum

The **real number** maximum expected value of the uncalibrated (original reversed) input. This is used for root selection when reversing higher order polynomials.

MultiPolynomial, MultiPoly, or MultiCal or MultiCalibration

Apply a series of interpolated calibrations to the data. The result is an interpolation of the two bounding calibrations for any point. This allows for a calibration to be adjusted over time, allowing for an approximation for slow drift between "true" calibrations.

Selection

The **data selection** to apply to.

Calibrations

A map keyed, on absolute time, of the **calibrations** to apply.

Extrapolate

A **boolean** instructing the action to extrapolate at the endpoints, instead of using the endpoint

calibration entirely when the time exceeds an endpoint.

Time

An **enumeration** controlling the time point each data value is evaluated with.

Middle **or** Center

Use the central time of the value.

Start **or** Begin

Use the start time of the value.

End

Use the end time of the value.

Interpolation

An **enumeration** specifying how the calibration results are interpolated together in time to produce the output.

Linear

Simple linear interpolation.

CubicSpline **or** CSpline

Cubic spline interpolation.

ClampStart

The **real number** slope of the interpolator at the first point. If set to undefined, a natural spline is used; meaning that the slope is set such that the second derivative is zero.

ClampEnd

The **real number** slope of the interpolator at the last point. If set to undefined, a natural spline is used; meaning that the slope is set such that the second derivative is zero.

Sin

Linear with a $\sin(x)$ based transfer function (concave).

SinPower

The **real number** power in the \sin^x . Larger values result in a steeper transition.

Cos

Linear with a $\cos(x)$ based transfer function (convex).

CosPower

The **real number** power in the \cos^x . Larger values result in a steeper transition.

Sigmoid

Linear with a sigmoid based transfer function (symmetric).

SigmoidSlope

The **real number** sharpness and shape of the sigmoid translation.

Wrap, Modular, **or** Modulus

Wrap values to within a constant range. For example, if a value exceeds the end it is wrapped around by the amount it exceeds it by from to the start. A common instance is a direction in degrees wrapped from zero to 360.

Selection

The [data selection](#) to apply wrapping to.

Start

The [real number](#) the lower start bound of the wrapping range.

End

The [real number](#) the upper end bound of the wrapping range.

Remove

Remove values entirely from the stream, creating a gap.

Selection

The [data selection](#) to remove.

Flavors

Alter the data identifier by adding and/or removing flavors from it.

Selection

The [data selection](#) before alteration.

ApplyToMetadata

A [boolean](#) controlling if alteration is also applied to metadata. When absent, the alteration is applied to metadata.

Add

An array of [text](#) flavor names to add to the identifier.

Remove

An array of [text](#) flavor names to remove from the identifier.

Unit **or** SetUnit

Alter the data identifier main components.

Selection

The [data selection](#) before alteration.

ApplyToMetadata

A [boolean](#) controlling if alteration is also applied to metadata. When absent, the alteration is applied to metadata.

Station

The [text](#) altered station identifier.

Archive

The [text](#) altered archive identifier.

Variable

The **text** altered variable identifier.

Flavors

An array of **text** altered identifier flavors.

UnitReplace **or** Translate

Alter the data identifier with substitutions extracted from regular expressions applied to the original. The regular expression captures are available in substations like "\${1}" for the first (non-whole pattern) capture on the associated component.

Selection

The **data selection** before alteration.

ApplyToMetadata

A **boolean** controlling if alteration is also applied to metadata. When absent, the alteration is applied to metadata.

From/Station

The **text** regular expression applied to the original station identifier.

From/Archive

The **text** regular expression applied to the original archive identifier.

From/Variable

The **text** regular expression applied to the original variable identifier.

From/Flavor

The **text** regular expression applied to the original flavors identifiers.

To/Station

The **text** altered station identifier.

To/Archive

The **text** altered archive identifier.

To/Variable

The **text** altered variable identifier.

To/Flavor

The **text** altered identifier flavor.

Duplicate

Create a duplicate stream of data by copying an existing one then altering its unique identifier.

Selection

The **data selection** before alteration.

ApplyToMetadata

A **boolean** controlling if alteration is also applied to metadata. When absent, the alteration is applied to metadata.

Station

The **text** altered station identifier.

Archive

The **text** altered archive identifier.

Variable

The **text** altered variable identifier.

Flavors

An array of **text** altered identifier flavors.

DuplicateTranslate

Create a duplicate stream of data by copying an existing one then altering its unique identifier. This variant also provides substitutions extracted from regular expressions applied to the original. The regular expression captures are available in substations like "\${1}" for the first (non-whole pattern) capture on the associated component.

Selection

The **data selection** before alteration.

ApplyToMetadata

A **boolean** controlling if alteration is also applied to metadata. When absent, the alteration is applied to metadata.

From/Station

The **text** regular expression applied to the original station identifier.

From/Archive

The **text** regular expression applied to the original archive identifier.

From/Variable

The **text** regular expression applied to the original variable identifier.

From/Flavor

The **text** regular expression applied to the original flavors identifiers.

To/Station

The **text** altered station identifier.

To/Archive

The **text** altered archive identifier.

To/Variable

The **text** altered variable identifier.

To/Flavor

The **text** altered identifier flavor.

Overlay **or** Set

Set a value directly by overlaying some contents at a specific path. If the overlay value is opaque then this is a simple set, while complex types are overlaid on any original contents.

Selection

The **data selection** to alter.

Path

The **path** within the original values to overlay at.

Value

The value contents to set the output to. Any time is allowed here and it is directly overlaid into the output.

Metadata **or** Meta **or** OverlayMeta **or** OverlayMetadata

Alter metadata by overlaying some contents at a specific path. If the overlay value is opaque then this is a simple set, while complex types are overlaid on any original contents.

Selection

The **data selection** to alter.

Path

The **path** within the original values to overlay at.

Value

The value contents to set the output to. Any time is allowed here and it is directly overlaid into the output.

Function, Math, **or** Arithmetic

Apply an arithmetic operation to the data. This allows for simple operations like adding one value to another one.

Selection

The **data selection** to alter. This also provides one of the inputs to the function being applied.

Inputs

The **data selection** of the other input. If multiple values are present, an unspecified valid one is selected.

Calibration

The **calibration** to applied to the pure input, before the function is applied.

Path

The **path** within the pure input values to read from.

Function

An **enumeration** specifying the function to apply to generate the output values.

Add **or** Plus

Add the input to the data (Selection + Calibration(Input)).

Subtract **or** Minus

Subtract the input from the data (Selection - Calibration(Input)).

Multiply **or** Times

Multiply the input and the data (Selection * Calibration(Input)).

Divide

Divide the data by the input (Selection / Calibration(Input)).

Invert **or** Inverse

Divide the input by the data, effectively inverting the data (Calibration(Input) / Selection).

Assign, Set, **or** Equal

Divide the input by the data, effectively inverting the data (Calibration(Input) / Selection).

ScriptDemultiplexer **or** Demultiplexer

Apply a script controlled segment demultiplexer to the data stream.

Code

The [script](#) code to execute to define the demultiplexer.

ScriptSegment, ScriptSegments, **or** Script

Apply a script processor for each segment in the data stream.

Code

The [script](#) code to execute for each time segment.

Selection

The [data selection](#) to apply the processor to. This serves as both the input and output to the processor if explicit ones are not defined.

Inputs

The [data selection](#) of explicit inputs.

Outputs

The [data selection](#) of explicit outputs.

ImplicitMetadata

A [boolean](#) controlling if metadata inputs are implicitly added to the segments. When absent, metadata are included.

ScriptValue **or** ScriptValues

Apply a script processor for each value in the data stream.

Code

The [script](#) code to execute for each value.

Selection

The [data selection](#) to use as inputs to the processor. All output values are placed into the data stream.

ScriptGeneralSegment **or** ScriptGeneralSegments

Execute a full script thread, with segments as the inputs to the data stream it sees.

Code

The **script** defining the execution thread. This normally includes a loop to process all inputs.

Selection

The **data selection** to apply to. This serves as both the input and output to the processor if explicit ones are not defined.

Inputs

The **data selection** of explicit inputs.

Outputs

The **data selection** of explicit outputs.

ImplicitMetadata

A **boolean** controlling if metadata inputs are implicitly added to the segments. When absent, metadata are included.

ScriptGeneralValue **or** ScriptGeneralValues

Execute a full script thread, with data values as the inputs to the data stream it sees.

Code

The **script** defining the execution thread. This normally includes a loop to process all inputs.

Selection

The **data selection** to use as inputs to the processor. All output values are placed into the data stream.

NOOP **or** None

No operation, effectively disabling the action.

Simple invalidation

```
/Type,"Invalidate"  
/Selection/#0/Variable,"BsG_S11"
```

```
/Type,"MultiSpotSize"  
/Instrument,"A11"  
/Original/#0,19.81  
/Original/#1,19.81  
/Original/#2,19.81  
/Original/#3,19.81  
/Original/#4,19.81  
/Original/#5,19.81  
/Original/#6,19.81  
/Original/#7,19.81  
/Corrected/#0,21.2  
/Corrected/#1,20.1  
/Corrected/#2,19.5  
/Corrected/#3,20.3  
/Corrected/#4,19.9  
/Corrected/#5,21.0  
/Corrected/#6,20.9  
/Corrected/#7,21.7
```

Editing Trigger

A trigger is a condition applied to a data stream resulting in a "match or no match" output stream. The most common case for this is edit directives, where the trigger governs if the action of the edit applies or not.

The triggering mechanism is also re-used elsewhere, but it always has the same effect: evaluating if a condition is met or not on data.

The trigger condition is selected with the [enumeration](#) value `Type`:

Always

The trigger is always met. This is the default when no type is selected.

Never **or** Disable

The trigger is never met.

HasFlag, HasFlags, Flag, **or** Flags

Test for the presence of all specified flags.

Flags

The [flags](#) to require.

Input

The [data selection](#) of the flags variable to test.

Path

The [path](#) in the selected variable to look for flags at.

Lacksflag **or** LacksFlags

Test for the absence of all specified flags.

Flags

The **flags** to require to be absent.

Input

The **data selection** of the flags variable to test.

Path

The **path** in the selected variable to look for flags at.

Periodic, Moment, **or** Instant

Activate the trigger at specific moments within a repeating interval in time. For example, a trigger that activates on the 15th, 30th, and 45th minutes after the hour.

Interval

The **time interval** governing the inspection range. Enabling alignment is usually required for repeatable results.

MomentUnit

The **enumeration** setting the **time interval** units that the moments are specified in.

Moments

An array of **integer** units after the interval start to activate on.

PeriodicRange **or** TimeRange

Activate the trigger for all data inside a range within a repeating interval in time. For example, a trigger that is active from 15 to 30 minutes after the hour.

Interval

The **time interval** governing the inspection range. Enabling alignment is usually required for repeatable results.

Start

The **time offset** after the beginning of the main interval to begin activation.

End

The **time offset** after the beginning of the main interval to end activation.

Exists

Activate the trigger when the specified data exists with any value, even undefined or invalid ones.

Input

The **data selection** of the value to test.

Path

The **path** within the values to inspect.

Defined

Activate the trigger when the specified data has a valid and non-missing value.

Input

The **data selection** of the value to test.

Undefined

Activate the trigger when the specified data exists but has an explicitly undefined value.

Input

The **data selection** of the value to test.

AND or All

Activate when all sub-triggers are also active, in the form of a logical AND operation.

Triggers

An array of further triggers to test.

OR or Any

Activate when any sub-trigger is active, in the form of a logical OR operation.

Triggers

An array of further triggers to test.

Less or Less than

Compare a left and right input, activating when the left one is strictly less than the right one.

Left

The **left hand** side of the inequality.

Right

The **right hand** side of the inequality.

LessEqual or LessThanOrEqual

Compare a left and right input, activating when the left one is less than or equal to the right one.

Left

The **left hand** side of the inequality.

Right

The **right hand** side of the inequality.

Greater or Greater Than

Compare a left and right input, activating when the left one is strictly greater than the right one.

Left

The **left hand** side of the inequality.

Right

The **right hand** side of the inequality.

GreaterEqual or GreaterThanOrEqual

Compare a left and right input, activating when the left one is greater than or equal to the right one.

Left

The **left hand** side of the inequality.

Right

The **right hand** side of the inequality.

Equal **or** EqualTo

Compare two inputs, activating when their values are equal.

Left

The first **input** to compare.

Right

The second **input** to compare.

NotEqual **or** NotEqualTo

Compare two inputs, activating when their values are not equal.

Left

The first **input** to compare.

Right

The second **input** to compare.

Range **or** InsideRange

Compare a value with a bound specification, activating when it is between the lower and upper bounds.

Start

The **start** of the comparison range.

End

The **end** of the comparison range.

Value

The **input** value to check.

ModularRange **or** InsideModularRange

Compare a value with a wrapped or modular bound specification, activating when it is between the lower and upper bounds. This also wraps the other inputs to a specific range, allowing the activation range to "pass through" the ends. For example, a specified as [0,360] with limits set to [350,10] activates on values of 355 and 5 but does not activate on 340.

Start

The **start** of the comparison range.

End

The **end** of the comparison range.

Value

The **input** value to check.

Modulus/Start

The **lower limit** of the total range.

Modulus/End

The **upper limit** of the total range.

OutsideRange

Compare a value with a bound specification, activating when it is outside the bounds defined by the lower and upper limits.

Start

The **start** of the comparison range.

End

The **end** of the comparison range.

Value

The **input** value to check.

OutsideModularRange

Compare a value with a wrapped or modular bound specification, activating when it is outside the bounds defined by the lower and upper limits. This also wraps the other inputs to a specific range, allowing the activation range to "pass through" the ends. For example, a specified as [0,360] with limits set to [350,10] activates on values of 340 and 20 but does not activate on 355 or 5.

Start

The **start** of the comparison range.

End

The **end** of the comparison range.

Value

The **input** value to check.

Modulus/Start

The **lower limit** of the total range.

Modulus/End

The **upper limit** of the total range.

Script

A trigger defined by evaluating script code for incoming time segments. The trigger is considered met when the script code returns true.

Code

The **script** code to execute for each segment.

Input

The [data selection](#) of inputs sent to the script.

In addition a trigger accepts several parameters controlling its output activation:

Invert

A [boolean](#) controlling if the trigger is considered met when the condition is false and unmet when it is true.

Extend

A [time interval](#) setting how far the trigger is considered to be met before and after the actual hit. For example, if set to five minutes the trigger will report that it was met for five minutes on both sides of a single "actual" instance of it being met.

Extend/Before

A [time interval](#) setting the extension before the actual hit only.

Extend/After

A [time interval](#) setting the extension after the actual hit only.

Always active

```
/Type, "Always"
```

Remove specific minutes after the hour

```
/Type, "Periodic"  
/Interval/Units, "Hour"  
/Interval/Count, 1  
/Interval/Align, TRUE  
/MomentUnit, "Minute"  
/Moments/#0, 0  
/Moments/#1, 15  
/Moments/#2, 30  
/Moments/#3, 45
```

Editing Trigger Numeric Inputs

Edit triggers can include numeric comparisons. To perform these the triggers provide an input conversion mechanism.

The input conversion allows for both data extraction and function conversion. Since this can be nested, this allows for complex function construction.

The input conversion is selected with the [enumeration](#) value `Type` within the input specification (e.g. `Left/Type`):

Value

A value read from the input data. This is the default when no type is selected.

Value

The **data selection** to read values from.

Path

The **path** in the selected variable to get the value at.

Constant

A constant value.

Value

The **real number** value of the input.

Additionally, if no type is selected and the entire trigger is a number, that number is intercepted as a constant value.

Sum **or** Add

Add together set of inputs to produce a final sum of their values.

RequireAll

The **boolean** indicating if all inputs are required to be valid to produce an output. When set explicitly to false, any invalid inputs are simply omitted from the final sum.

Inputs

An array of **inputs** to sum together.

Difference **or** Subtract

Subtract inputs sequentially from the first one.

RequireAll

The **boolean** indicating if all inputs are required to be valid to produce an output. When set explicitly to false, any invalid inputs are simply omitted from the final difference.

Inputs

An array of **inputs** to subtract. The first valid one is the initial value and further ones are subtracted from it (First - Second - Third ...).

Product **or** Multiply

Multiply together set of inputs to produce a final product of their values.

RequireAll

The **boolean** indicating if all inputs are required to be valid to produce an output. When set explicitly to false, any invalid inputs are simply omitted from the final product.

Inputs

An array of **inputs** to multiply together.

Quotient **or** Divide

Divide inputs to produce a final quotient value.

RequireAll

The **boolean** indicating if all inputs are required to be valid to produce an output. When set

explicitly to false, any invalid inputs are simply omitted from the final value.

Inputs

An array of **inputs** to divide. The first valid input is the initial value and all subsequent ones are subsequent ones are divided from it (First / Second / Third ...).

Power

Raise inputs to powers to produce a final value.

RequireAll

The **boolean** indicating if all inputs are required to be valid to produce an output. When set explicitly to false, any invalid inputs are simply omitted from the final value.

Inputs

An array of **inputs** to raise to each others power. The first valid input is the initial value with subsequent ones being the powers the previous value is raised to ((First ^ Second) ^ Third ...).

Largest

Select the largest value in a set of inputs. The final output value is the value of the largest (closest to +infinity) of the inputs.

RequireAll

The **boolean** indicating if all inputs are required to be valid to produce an output. When set explicitly to false, any invalid inputs are simply ignored for the calculation.

Inputs

An array of **inputs** to inspect and select from.

Smallest

Select the smallest value in a set of inputs. The final output value is the value of the largest (closest to -infinity) of the inputs.

RequireAll

The **boolean** indicating if all inputs are required to be valid to produce an output. When set explicitly to false, any invalid inputs are simply ignored for the calculation.

Inputs

An array of **inputs** to inspect and select from.

First, FirstValid, or Valid

Select the first valid and defined input from a list.

Inputs

An array of **inputs** to inspect and select from. The first one in the order of the array that has a valid and defined value is the final value of the input.

Polynomial, Calibration, Poly, or Cal

Apply a calibration value to an input value.

Input

The **input** value to the calibration.

Calibration

The **calibration** applied to the input.

PolynomialInvert, PolyInvert, or InvertCal

Invert or back out a calibration from an input value. The result is the value before the calibration was applied.

Input

The **input** value to the calibration inversion.

Calibration

The **calibration** that is inverted.

Minimum

The **real number** minimum expected value of the uncalibrated (original reversed) input. This is used for root selection when reversing higher order polynomials.

Maximum

The **real number** maximum expected value of the uncalibrated (original reversed) input. This is used for root selection when reversing higher order polynomials.

sin

Apply $\sin(x)$ to an input value.

Input

The **input** to the function.

cos

Apply $\cos(x)$ to an input value.

Input

The **input** to the function.

log or ln

Apply $\ln(x)$ to an input value.

Input

The **input** to the function.

log10

Apply $\log_{10}(x)$ to an input value.

Input

The **input** to the function.

exp

Apply e^x to an input value.

Input

The **input** to the function.

`abs`, `Absolute`, **or** `AbsoluteValue`

Take the absolute value of an input.

Input

The **input** to the function.

Mean

Calculate the arithmetic mean of another input over some moving window of time. The value is centered at a given time of the input, with the inspected data extending backwards and forwards from that time.

Input

The **input** to inspect.

Interval

The **time interval** to inspect on both sides of the current time.

Before

The **time interval** to inspect before the current time. The value takes precedence over the symmetric one, if both are present.

After

The **time interval** to inspect after the current time. The value takes precedence over the symmetric one, if both are present.

Gap

The **time interval** governing the gap threshold. If there is a gap more than this amount of time between two values, the input buffer is split and considered discontinuous.

`StandardDeviation` **or** `sd`

Calculate the standard deviation of another input over some moving window of time. The value is centered at a given time of the input, with the inspected data extending backwards and forwards from that time.

Input

The **input** to inspect.

Interval

The **time interval** to inspect on both sides of the current time.

Before

The **time interval** to inspect before the current time. The value takes precedence over the symmetric one, if both are present.

After

The **time interval** to inspect after the current time. The value takes precedence over the symmetric one, if both are present.

Gap

The **time interval** governing the gap threshold. If there is a gap more than this amount of time between two values, the input buffer is split and considered discontinuous.

Quantile

Calculate a quantile of another input over some moving window of time. The value is centered at a given time of the input, with the inspected data extending backwards and forwards from that time.

Input

The **input** to inspect.

Quantile

The **real number** quantile to calculate, in the range [0,1].

Interval

The **time interval** to inspect on both sides of the current time.

Before

The **time interval** to inspect before the current time. The value takes precedence over the symmetric one, if both are present.

After

The **time interval** to inspect after the current time. The value takes precedence over the symmetric one, if both are present.

Gap

The **time interval** governing the gap threshold. If there is a gap more than this amount of time between two values, the input buffer is split and considered discontinuous.

Median

Calculate the median of another input over some moving window of time. The value is centered at a given time of the input, with the inspected data extending backwards and forwards from that time.

Input

The **input** to inspect.

Interval

The **time interval** to inspect on both sides of the current time.

Before

The **time interval** to inspect before the current time. The value takes precedence over the symmetric one, if both are present.

After

The **time interval** to inspect after the current time. The value takes precedence over the symmetric one, if both are present.

Gap

The **time interval** governing the gap threshold. If there is a gap more than this amount of time between two values, the input buffer is split and considered discontinuous.

Minimum **or** min

Calculate the minimum (closest to -infinity) of another input over some moving window of time.

The value is centered at a given time of the input, with the inspected data extending backwards and forwards from that time. The result is the minimum value of all values being considered.

Input

The **input** to inspect.

Interval

The **time interval** to inspect on both sides of the current time.

Before

The **time interval** to inspect before the current time. The value takes precedence over the symmetric one, if both are present.

After

The **time interval** to inspect after the current time. The value takes precedence over the symmetric one, if both are present.

Gap

The **time interval** governing the gap threshold. If there is a gap more than this amount of time between two values, the input buffer is split and considered discontinuous.

Maximum **or** max

Calculate the maximum (closest to +infinity) of another input over some moving window of time. The value is centered at a given time of the input, with the inspected data extending backwards and forwards from that time. The result is the maximum value of all values being considered.

Input

The **input** to inspect.

Interval

The **time interval** to inspect on both sides of the current time.

Before

The **time interval** to inspect before the current time. The value takes precedence over the symmetric one, if both are present.

After

The **time interval** to inspect after the current time. The value takes precedence over the symmetric one, if both are present.

Gap

The **time interval** governing the gap threshold. If there is a gap more than this amount of time between two values, the input buffer is split and considered discontinuous.

Slope

Calculate the slope of another input over some moving window of time based on linear regression in seconds. The value is centered at a given time of the input, with the inspected data extending backwards and forwards from that time. The result is the slope of a least squares regression fit with the X dimension being the time in seconds. So the final output value is the input's units per second.

Input

The **input** to inspect.

Interval

The **time interval** to inspect on both sides of the current time.

Before

The **time interval** to inspect before the current time. The value takes precedence over the symmetric one, if both are present.

After

The **time interval** to inspect after the current time. The value takes precedence over the symmetric one, if both are present.

Gap

The **time interval** governing the gap threshold. If there is a gap more than this amount of time between two values, the input buffer is split and considered discontinuous.

Elapsed, Length, or Duration

Calculate the amount of time in a moving window of time. The value is centered at a given time of the input, with the inspected data extending backwards and forwards from that time. The result is simply the end of the final value being considered minus the first value, in seconds.

Input

The **input** to inspect.

Interval

The **time interval** to inspect on both sides of the current time.

Before

The **time interval** to inspect before the current time. The value takes precedence over the symmetric one, if both are present.

After

The **time interval** to inspect after the current time. The value takes precedence over the symmetric one, if both are present.

Gap

The **time interval** governing the gap threshold. If there is a gap more than this amount of time between two values, the input buffer is split and considered discontinuous.

Average or Smoothed

Calculate the correctly weighted arithmetic mean of an input in a moving window of time. The value is centered at a given time of the input, with the inspected data extending backwards and forwards from that time. The result is the mean value with coverage and time contributions being considered. This allows for emulation of the final normal averaging applied to data.

Value

The **data selection** to read values from.

Path

The **path** in the selected variable to get the value at.

Interval

The **time interval** to inspect on both sides of the current time.

Before

The **time interval** to inspect before the current time. The value takes precedence over the symmetric one, if both are present.

After

The **time interval** to inspect after the current time. The value takes precedence over the symmetric one, if both are present.

Gap

The **time interval** governing the gap threshold. If there is a gap more than this amount of time between two values, the input buffer is split and considered discontinuous.

Symmetric extended threshold

```
/Type,"GreaterThan"  
/Left/Value/#0/Variable,"BsG_S11"  
/Right,5.0  
/Extend/Units,"Minute"  
/Extend/Count,5
```

Asymmetric risk threshold

```
/Type,"GreaterThan"  
/Left/Value/#0/Variable,"BsG_S11"  
/Extend/Before/Units,"Second"  
/Extend/Before/Count,30  
/Right/Type,"Product"  
/Right/Inputs/#0,2.5  
/Right/Inputs/#1/Type,"Mean"  
/Right/Inputs/#1/Interval/Units,"Minute"  
/Right/Inputs/#1/Interval/Count,15  
/Right/Inputs/#1/Input/Type,"Value"  
/Right/Inputs/#1/Input/Value/Variable,"BsG_S11"
```

Examples

Minimal Acquisition System

In some cases, no configuration may be required at all. That is, if the acquisition system does not need fixed instruments or doesn't need them to be assigned fixed identifiers, the entirely automatic auto-configuration can be used.

However, in most cases a minimal configuration is required. This provides a basic definition of the instruments connected to the system. The below example only contains a single instrument, but it can be extended to others by changing the unique identifier (the **MCPC** below).

```
/aerosol/Autoprobe/DisableDefaultComponents,TRUE ①  
/aerosol/Components/MCPC/Instrument,"N61" ②  
/aerosol/Components/MCPC/Name,"acquire_bmi_cpc1720" ③
```

- ① This line disables the default components involved in auto-detection. This means that only components explicitly listed will be considered, so all possible instruments on the system must also be defined.
- ② This line sets the instrument code explicitly. Since all instruments are going to be defined, it makes sense to also set their instrument codes to fixed identifiers.
- ③ The final line sets the internal component name the instrument is using. Since the default component are disabled, an explicitly listed component is required. This component controls how the the system interacts over the serial port.

Manual Serial Number

For many instruments, the serial number and instrument identification can be read automatically. When it cannot be, it must then be specified manually. The most common example of this is the TSI 3563 Nephelometer, which is used as the below example.

```
/aerosol/Autoprobe/DisableDefaultComponents,TRUE  
/aerosol/Components/Neph/Name,"acquire_tsi_neph3563"  
/aerosol/Components/Neph/Source/SerialNumber,1087 ①
```

- ① This line sets the serial number added to all metadata generated by the component. Though text is allowed, if possible integer numbers are preferred, stripping out the boilerplate parts of the serial number that are constant between all instruments.

Manual Metadata

In some cases it makes sense to track additional metadata for variables generated by an instrument. The below case shows how to add metadata for a specific variable.

```
/aerosol/Components/CPC/Flow,1.5396 ①  
/aerosol/Components/CPC/Groups,|cpc ②  
/aerosol/Components/CPC/Instrument,"N61"  
/aerosol/Components/CPC/MenuCharacter,"C" ③  
/aerosol/Components/CPC/Name,"acquire_gmd_cpcpulse"  
/aerosol/Components/CPC/Source/Manufacturer,"TSI" ④  
/aerosol/Components/CPC/Source/Model,"3760"  
/aerosol/Components/CPC/Source/SerialNumber,327  
/aerosol/Components/CPC/VariableMetadata/N_.*/*dNoteFlow,"150123/PJS@GMD" ⑤
```

- ① Here an instrument specific parameter is set. In this case, this is the flow rate in LPM used to convert the count rate into a number concentration.
- ② This line assigns the instrument to specific groups for control, which comes up later when the system needs to control different sets of instruments. For now, in this simple configuration, this does nothing.
- ③ This line sets a menu character for the instrument. This is the hot key on the realtime display menus, this can normally be left absent unless there are multiple instances of the same class of instrument. In that scenario it is often a good idea to specify so they get assigned to fixed constant ones.
- ④ As above, these lines set instrument specific metadata attached to all variables. In this case, the manufacturer and model are required, because the source component is a generic one not tied to a specific model of instrument.
- ⑤ Finally, the actual metadata specific line. The `N_.*` is the regular expression matching the variables to apply to. In this case, it matches anything starting with `N_` meaning the concentration generated. The `*dNoteFlow` specifies the actual metadata path. Here it is a the key `NoteFlow` specific to real values.

Matched Metadata

Instruments that report disambiguation metadata can be used with a match entry to restrict the component to only apply to a specific physical instance. The common case for this is if a calibration is set in the configuration that should only be written to a specific instance of an instrument.

```

/aerosol/Components/CLAP/Area/#0,19.74
/aerosol/Components/CLAP/Area/#1,19.71
/aerosol/Components/CLAP/Area/#2,20.05
/aerosol/Components/CLAP/Area/#3,19.25
/aerosol/Components/CLAP/Area/#4,19.76
/aerosol/Components/CLAP/Area/#5,19.65
/aerosol/Components/CLAP/Area/#6,19.47
/aerosol/Components/CLAP/Area/#7,19.88
/aerosol/Components/CLAP/FlowScale,1.0
/aerosol/Components/CLAP/Groups,sample|spancheck ①
/aerosol/Components/CLAP/HardwareFlowCalibration/#0,-0.13997
/aerosol/Components/CLAP/HardwareFlowCalibration/#1,0.0717
/aerosol/Components/CLAP/HardwareFlowCalibration/#2,0.04249
/aerosol/Components/CLAP/HardwareFlowCalibration/#3,0.01586
/aerosol/Components/CLAP/Instrument,"A11"
/aerosol/Components/CLAP/Match/SerialNumber,9 ②
/aerosol/Components/CLAP/Name,"acquire_gmd_clap3w"
/aerosol/Components/CLAP/VariableMetadata/Q_.*/*dNoteCalibration,"10.009#2015-03-20/PJS@BND"
/aerosol/Components/CLAP/VariableMetadata/Q_.*/*dNoteScale,"10.009#2015-03-20/PJS@BND"

```

- ① This line assigned the instrument to multiple control groups. Anything affecting either control group also affects this instrument.
- ② This line sets the match required. The component will not be instantiated if it fails to match. The data here must match the generated metadata exactly (see the "Source" section of generated variables). Only the given keys here are required to exist and match, any additional ones in the metadata are ignored.

Fixed Control Interface

When an instrument cannot be auto-detected or disambiguated from another instance of it, a fixed interface can be used. This allows you to set the specific serial port the instrument is connected to, for example. However, this should be avoided, if possible, because it limits the ability of the system to compensate for unreliable hardware.

```

/aerosol/Autoprobe/DisableDefaultConfiguration,TRUE ①
/aerosol/Autoprobe/DisableDefaultInterfaces,TRUE ②
/aerosol/Components/N17/Instrument,"N17"
/aerosol/Components/N17/Interface/Port,"/dev/ttyUSB6" ③
/aerosol/Components/N17/Interface/Type,"SerialPort" ④

```

- ① This example also demonstrates how to define instruments that are allowed to match multiple possible types. This line tells the system to only use the defined configurations of instruments, rather than allowing the default configured ones as well.
- ② This line tells the system not to use any of the default system detected interfaces (serial ports). When only a part of the configuration is using fixed interfaces, this line must be omitted.
- ③

This line sets the port for the N17 instrument to `/dev/ttyUSB6` (the seventh USB to serial port converter, on Linux).

④ This line sets the type of communications interface in use.

Generic Conversion

Some types of instruments do not automatically define variables for all their inputs. For example, all the forms of generic analog to digital converters. These instruments require definitions of the variables they generate from their raw sampling channels.

```
/aerosol/Components/uMAC/DigitalOutputs/DisableBypassValve,0 ①
/aerosol/Components/uMAC/DigitalOutputs/PM1Impactor,1
/aerosol/Components/uMAC/Bypass/Digital/DisableBypassValve,FALSE ②
/aerosol/Components/uMAC/Exit/Digital/DisableBypassValve,FALSE
/aerosol/Components/uMAC/Initialize/Digital/DisableBypassValve,TRUE
/aerosol/Components/uMAC/Groups,control|spancheck
/aerosol/Components/uMAC/Instrument,"X1"
/aerosol/Components/uMAC/Name,"acquire_azonix_umac1050"
/aerosol/Components/uMAC/UnBypass/Digital/DisableBypassValve,TRUE
/aerosol/Components/uMAC/VariableGroups/#0/Groups,|sample ③
/aerosol/Components/uMAC/VariableGroups/#0/Variable,"T_V11"
/aerosol/Components/uMAC/VariableGroups/#1/Groups,|sample
/aerosol/Components/uMAC/VariableGroups/#1/Variable,"Pd_P11"
/aerosol/Components/uMAC/VariableGroups/#4/Groups,|control
/aerosol/Components/uMAC/VariableGroups/#4/Variable,"(T|(Pd?)|U|Q)_(T|P|V|Q)[560][0-9]"
/aerosol/Components/uMAC/Variables/Pd_P01/Channel,19 ④
/aerosol/Components/uMAC/Variables/Pd_P01/Calibration/#0,-0.8193 ⑤
/aerosol/Components/uMAC/Variables/Pd_P01/Calibration/#1,4.1192
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dDescription,"Stack pitot tube" ⑥
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dFormat,"00.000"
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dGroupUnits,"dhPa"
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dNoteCalibration,"121014/PJS@BND#-0.743;4.012"
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dUnits,"hPa"
/aerosol/Components/uMAC/Variables/Pd_P11/Calibration/#0,-209.15
/aerosol/Components/uMAC/Variables/Pd_P11/Calibration/#1,338.15
/aerosol/Components/uMAC/Variables/Pd_P11/Channel,21
/aerosol/Components/uMAC/Variables/Pd_P11/Metadata/*dDescription,"Impactor pressure drop"
/aerosol/Components/uMAC/Variables/Pd_P11/Metadata/*dGroupUnits,"dhPa"
/aerosol/Components/uMAC/Variables/Pd_P11/Metadata/*dNoteCalibration,"#150318/PJS@BND#-210.64;338.82#121014/PJS@BND#-203.6;332.13#110808/PJS@BND#-202.71;325.15#101201/PJS@BND#-212.02;333.33"
/aerosol/Components/uMAC/Variables/Pd_P11/Metadata/*dUnits,"hPa"
```

① In addition to pure analog inputs, the μ MAC provides a number of digital outputs. This line defines channel zero with the name `DisableBypassValve`. That name can then be used by various control processes to manipulate the associated digital channel, without requiring them to know

the exact assignment.

- ② Once a digital channel has been defined, the μ MAC is also capable of controlling it on its own for some simple conditions. This line indicates that the channel should be set to low when the system enters into bypass mode.
- ③ Control groups for variables determine how they are averaged and cut split. Any instrument can assign its resulting variables to different control groups (they default to the instrument global ones). However, this is normally only required for generic ones that may span different parts of the system. In this example, the μ MAC itself is not part of the `sample` control group, which is what normally determines the cut size. Since the `T_V11` variable is split by cut size, it must be placed into that control group so the system size selection applies to it
- ④ This line defines the variable `Pd_P01` attached to analog input channel number 19.
- ⑤ Since the μ MAC reads its inputs in raw voltages, a calibration is required to convert to physical units.
- ⑥ These lines set the base metadata for the specific variable. Generic defined variables should provide the basic set of metadata to describe the variable to the system.

Simple Scheduling

When a continuously cycling schedule is required, the `control_cycle` component is used to define a set of actions executed at various points within the greater interval. This example is a simple hourly zero.

```
/aerosol/Components/ZeroSchedule/Actions/#0/Unit,"second" ①  
/aerosol/Components/ZeroSchedule/Actions/#0/Count,3418  
/aerosol/Components/ZeroSchedule/Actions/#0/Aligned,TRUE  
/aerosol/Components/ZeroSchedule/Actions/#0/Commands/StartZero,TRUE ②  
/aerosol/Components/ZeroSchedule/Actions/#0/Description,"System Zero" ③  
/aerosol/Components/ZeroSchedule/Fixed,TRUE ④  
/aerosol/Components/ZeroSchedule/Groups,|sample  
/aerosol/Components/ZeroSchedule/MenuCharacter,""  
/aerosol/Components/ZeroSchedule/MenuEntry,"Zero Schedule" ⑤  
/aerosol/Components/ZeroSchedule/Name,"control_cycle"
```

- ① Every entry within the action list is based off a `time interval` definition. So, these lines define the point within the total time period to execute the action at. The total time interval defaults to one hour aligned, so this action is executed at 3418 seconds after the hour.
- ② The action then contains the various operations it performs on execution. Here, the only action is to set the `StartZero` command to all instruments matching the output control groups. Since the schedule itself belongs to the `sample` control group, only other instruments also in that group receive the command.
- ③ This line sets a human readable description of the action. This is used to generate menu entries and in other locations where the automatically generated one is not clear.
- ④ This line tells the acquisition system to always create the component and to ignore all auto-detection. Since control components cannot be auto-detected, this is required for all of them.

⑤ This line sets the menu entry and window title for the controller.

Simple Triggering

To generate actions based on if a trigger condition is met, the `control_trigger` component is used. This example bypasses the system when the the impactor pressure exceeds a threshold then requires manual user interaction before un-bypassing.

```
/aerosol/Components/BypassFailsafe/Actions/#0/Bypass,Bypass|ImpactorPressureTooHigh ①
/aerosol/Components/BypassFailsafe/Actions/#0/Continuous,TRUE ②
/aerosol/Components/BypassFailsafe/Actions/#0/Dependencies/Acquire,|ImpactorPressureBy
pass ③
/aerosol/Components/BypassFailsafe/Actions/#0/Dependencies/Hold,30.0 ④
/aerosol/Components/BypassFailsafe/Actions/#0/Event,"Pressure across the impactor is
too high. System bypassed. Please check the manual ball valve." ⑤
/aerosol/Components/BypassFailsafe/Actions/#0/EventRealtime,TRUE ⑥
/aerosol/Components/BypassFailsafe/Actions/#0/Triggers/#0/Input,"rt_instant:Pd_P11" ⑦
/aerosol/Components/BypassFailsafe/Actions/#0/Triggers/#0/Threshold,100.0 ⑧
/aerosol/Components/BypassFailsafe/Actions/#0/Triggers/#0/Type,"Threshold"
/aerosol/Components/BypassFailsafe/Actions/#1/ClearBypass,|ImpactorPressureTooHigh
/aerosol/Components/BypassFailsafe/Actions/#1/Dependencies/Release,|ImpactorPressureBy
pass ⑨
/aerosol/Components/BypassFailsafe/Actions/#1/Event,"Pressure across the impactor
returned to normal. The system lock has been released but a manual release is
required before sampling resumes."
/aerosol/Components/BypassFailsafe/Actions/#1/EventRealtime,TRUE
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Input,"rt_instant:Pd_P11"
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Operation,"Less"
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Require,|ImpactorPressureBy
pass ⑩
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Threshold,90.0
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Type,"Threshold"
/aerosol/Components/BypassFailsafe/Fixed,TRUE
/aerosol/Components/BypassFailsafe/Groups,control|sample
/aerosol/Components/BypassFailsafe/MenuCharacter,""
/aerosol/Components/BypassFailsafe/MenuEntry,"Bypass Failsafe"
/aerosol/Components/BypassFailsafe/Name,"control_trigger"
```

① This line indicates that the action bypasses the system. Here two bypass flags are used, the first `Bypass` is a global indicator that is normally only control by user interaction. The second is flag that is released by the second action when the pressure returns to normal.

② This line indicates that the action should continue to execute as long as the trigger is met, instead of only once when it is first met. Here this is used to ensure the system stays bypassed until the pressure is normal again.

③ This line defines a dependency tag that the action sets when it executed. Dependency tags are used by actions to allow for some of them to only execute after another has run.

④ The dependency hold line indicates to the controller that all the dependencies of the action

should be considered in limbo for 30 seconds even after the pressure drops below 100. This ensures that the bypass is always held for at least 30 seconds.

- ⑤ This line sets the message logged by the system when the action first executes.
- ⑥ This line indicates that the message is considered of realtime importance and is shown to the user by default.
- ⑦ Each action has one or more triggers that control its execution. Here the input to the trigger is set to the variable `Pd_P11` from the `rt_instant` archive. This indicates the impactor pressure from the instantaneous (un-averaged) feed.
- ⑧ This line sets the threshold used by the trigger.
- ⑨ This line indicates that once the action is executed (the pressure returns to normal), the dependency tag should be released. This means that any actions that depend on that tag will not execute again until it is set. For this context, this prevents the execution of the "return to normal" action until after the "abnormal" one has executed again.
- ⑩ Each trigger may also require dependency tags. Here this is used to make sure the action does not execute until after the pressure has gone abnormal.

Carousel Accumulation

To handle volume accumulation of a sampling carousel, the `control_accumulate` component is used. This example is the standard chemical filter sampling carousel at Barrow.

```

/aerosol/Components/FilterAccumulator/Accumulators/#0/Metadata/*dDescription,"Filter
bypass volume"
/aerosol/Components/FilterAccumulator/Accumulators/#0/Metadata/*dReportP,1013.25
/aerosol/Components/FilterAccumulator/Accumulators/#0/Metadata/*dReportT,0.0
/aerosol/Components/FilterAccumulator/Accumulators/#0/Metadata/*dUnits,"m³"
/aerosol/Components/FilterAccumulator/Accumulators/#0/Name,"Qt0" ①
/aerosol/Components/FilterAccumulator/Accumulators/#0/Rate/Input,"Q_Q21" ②
/aerosol/Components/FilterAccumulator/Accumulators/#0/Rate/Calibration/#0,0.0
/aerosol/Components/FilterAccumulator/Accumulators/#0/Rate/Calibration/#1,1.66667e-05

...

/aerosol/Components/FilterAccumulator/Accumulators/#8/Activate/InstrumentCommands/X3/S
etDigital/FilterValve8,TRUE ③
/aerosol/Components/FilterAccumulator/Accumulators/#8/Deactivate/InstrumentCommands/X3
/SetDigital/FilterValve8,FALSE
/aerosol/Components/FilterAccumulator/Accumulators/#8/Name,"Qt8"
/aerosol/Components/FilterAccumulator/Accumulators/#8/Rate/Calibration/#0,0.0
/aerosol/Components/FilterAccumulator/Accumulators/#8/Rate/Calibration/#1,1.66667e-05
/aerosol/Components/FilterAccumulator/Accumulators/#8/Rate/Input,"Q_Q21"
/aerosol/Components/FilterAccumulator/Advance/Unit,"Day" ④
/aerosol/Components/FilterAccumulator/Advance/Count,1
/aerosol/Components/FilterAccumulator/Advance/Align,TRUE
/aerosol/Components/FilterAccumulator/BlankIndex,1 ⑤
/aerosol/Components/FilterAccumulator/BlankTime/Unit,"Second" ⑥
/aerosol/Components/FilterAccumulator/BlankTime/Count,10
/aerosol/Components/FilterAccumulator/BypassIndex,0 ⑦
/aerosol/Components/FilterAccumulator/BypassOnEnd,TRUE ⑧
/aerosol/Components/FilterAccumulator/BypassWhileChanging,TRUE ⑨
/aerosol/Components/FilterAccumulator/Fixed,TRUE
/aerosol/Components/FilterAccumulator/Groups,|filter
/aerosol/Components/FilterAccumulator/Instrument,"F21"
/aerosol/Components/FilterAccumulator/MenuCharacter,""
/aerosol/Components/FilterAccumulator/MenuEntry,"Filter Accumulation Control"
/aerosol/Components/FilterAccumulator/MenuHide,TRUE
/aerosol/Components/FilterAccumulator/Name,"control_accumulate"
/aerosol/Components/FilterAccumulator/OutputGroups,|filterbypass ⑩
/aerosol/Components/FilterAccumulator/StartIndex,2 ⑪

```

- ① As with generic input variables, accumulator variables require a general name. Since they are defined as an array, they cannot derive their name from the definition path, so an explicit one is required.
- ② Every accumulation variable requires an input defining the rate in units/second to add to it. Here the input comes from the flow `Q_Q21` with a calibration to convert from LPM to m³/s, which is then integrated into just m³.
- ③ Each accumulator bin can define operations taken when the bin is activated or deactivated. In general, these should be the actions required to start and stop the accumulation through the bin. Here, the commands open and close the specific valve associated with the filter on the rack.

- ④ The time between accumulator advances. That is, the time to sample each accumulator before moving to the next one.
- ⑤ The accumulation bin to blank on, before starting the normal sampling rotation.
- ⑥ The time to blank before starting sampling.
- ⑦ The index to switch to when the system enters into bypass mode.
- ⑧ This line indicates that the accumulator should put the system into bypass when all accumulators are done sampling.
- ⑨ Similarly, this line indicates that the system should be bypassed when changing the accumulator.
- ⑩ This line sets the output control groups that the component sends to by default. In this example, this means that bypass flag set on carousel completion is only sent to the `filterbypass` group rather than the default `filter`. This is used to prevent the entire control group from bypassing when the filter completes.
- ⑪ This line sets the initial sampling index, when not blanking or bypassed.

Generic Input

For some simple instruments that only output fixed records without any need for control, it is sometimes possible to use the `acquire_generic_passive` component to interpret their data. The below example is the usage of the realtime wind feed at Barrow.

```

/aerosol/Components/RealtimeWind/AllowUnmatchedLines,TRUE ①
/aerosol/Components/RealtimeWind/Fixed,TRUE
/aerosol/Components/RealtimeWind/Instrument,"XM2"
/aerosol/Components/RealtimeWind/InstrumentMetadata/Manufacturer,"GMD"
/aerosol/Components/RealtimeWind/Interface/Baud,1200
/aerosol/Components/RealtimeWind/Interface/Port,"/dev/ttyUSB14"
/aerosol/Components/RealtimeWind/Interface/Type,"SerialPort"
/aerosol/Components/RealtimeWind/MenuCharacter,"W"
/aerosol/Components/RealtimeWind/MenuEntry,"NOAA Windbird"
/aerosol/Components/RealtimeWind/Name,"acquire_generic_passive"
/aerosol/Components/RealtimeWind/Records/#0/Match,"\\s*\\d+.*" ②
/aerosol/Components/RealtimeWind/Records/#0/Variables/WD/Calibration/#0,-180
/aerosol/Components/RealtimeWind/Records/#0/Variables/WD/Calibration/#1,1.0
/aerosol/Components/RealtimeWind/Records/#0/Variables/WD/Fields/#0,1 ③
/aerosol/Components/RealtimeWind/Records/#0/Variables/WD/Metadata/*dDescription,"Wind
direction from true north"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WD/Metadata/*dFormat,"000.0"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WD/Metadata/*dSmoothing/Mode,"Ve
ctor2D"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WD/Metadata/*dSmoothing/Paramete
rs/Direction,"WD"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WD/Metadata/*dSmoothing/Paramete
rs/Magnitude,"WS"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WD/Metadata/*dUnits,"degrees"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WS/Fields/#0,2
/aerosol/Components/RealtimeWind/Records/#0/Variables/WS/MaximumAge,30.0 ④
/aerosol/Components/RealtimeWind/Records/#0/Variables/WS/Metadata/*dDescription,"Wind
speed"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WS/Metadata/*dFormat,"00.0"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WS/Metadata/*dSmoothing/Mode,"Ve
ctor2D"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WS/Metadata/*dSmoothing/Paramete
rs/Direction,"WD"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WS/Metadata/*dSmoothing/Paramete
rs/Magnitude,"WS"
/aerosol/Components/RealtimeWind/Records/#0/Variables/WS/Metadata/*dUnits,"m/s"
/aerosol/Components/RealtimeWind/Timeout,30.0 ⑤

```

- ① This line allows the generic component to accept lines not matching any defined record without dropping communications.
- ② This line sets the match pattern for the defined record. This is a regular expression that must match the whole line for the record to be accepted. Here it simply looks for any line starting with a number.
- ③ As with generic definitions above, the component requires definitions of each output variable as a function of fields (CSV by default) in the record.
- ④ This line sets the maximum time between defined values before a gap in this variable is generated. This is required here because the realtime feed sometimes omits this field. So rather than having multiple gaps in a minute, this allows the gap to simply continue with the last

known value.

- ⑤ This line sets the total timeout with no recognized records before communications are dropped.

Realtime Export

The system also provides a simple component that can be used to generate basic output files directly from the realtime data stream. This is normally used to provide data to an alternate acquisition system as it is acquired. The most basic usage of this is simply generating a CSV file with a few columns written to a local disk. The below example is the output used at UGR to provide a feed to their existing plotting system.

```
/aerosol/Components/RealtimeDataFile/Columns/#0/TimeFormat,"yyyy-dd-MM hh:mm" ①
/aerosol/Components/RealtimeDataFile/Columns/#0/Type,"Time"
/aerosol/Components/RealtimeDataFile/Columns/#1/Calibration/#0,0.0
/aerosol/Components/RealtimeDataFile/Columns/#1/Calibration/#1,0.333333
/aerosol/Components/RealtimeDataFile/Columns/#1/Format,"0000.0"
/aerosol/Components/RealtimeDataFile/Columns/#1/Input/Variable,"BsG_S11" ②
/aerosol/Components/RealtimeDataFile/Columns/#1/MVC,"9999.9"
/aerosol/Components/RealtimeDataFile/Columns/#1/Type,"Value"
/aerosol/Components/RealtimeDataFile/Columns/#2/Format,"0000.0"
/aerosol/Components/RealtimeDataFile/Columns/#2/Input/Variable,"XR_A31" ③
/aerosol/Components/RealtimeDataFile/Columns/#2/MVC,"9999.9"
/aerosol/Components/RealtimeDataFile/Columns/#2/Type,"Value"
/aerosol/Components/RealtimeDataFile/Command,"curl -u XXXX:YYYY -T ${FILE} -m 30 -s
--ftp-pasv --disable-epsv --ftp-create-dirs
ftp://some.ftp.server.edu/pm_ebc/${TIME|FORMAT|yyyyddMM}.csv" ④
/aerosol/Components/RealtimeDataFile/Duration/Count,1 ⑤
/aerosol/Components/RealtimeDataFile/Duration/Unit,"Day"
/aerosol/Components/RealtimeDataFile/Duration/Align,TRUE
/aerosol/Components/RealtimeDataFile/Fixed,TRUE
/aerosol/Components/RealtimeDataFile/Header,FALSE ⑥
/aerosol/Components/RealtimeDataFile/LoadArchived,TRUE ⑦
/aerosol/Components/RealtimeDataFile/MenuCharacter,""
/aerosol/Components/RealtimeDataFile/MenuEntry,"Realtime Data Uploader"
/aerosol/Components/RealtimeDataFile/MenuHide,TRUE ⑧
/aerosol/Components/RealtimeDataFile/Name,"control_export"
```

- ① The first column is a simple time formatted column.
- ② The second column is the nephelometer green scattering, scaled by 1/3 (to a mass concentration).
- ③ The third column is the EBC from the MAAP.
- ④ This is a command run every time the export file is updated. This instance uploads the file to a FTP server, using the username "XXXX" and the password "YYYY".
- ⑤ This sets the total duration of the export file, back from the latest received value.
- ⑥ This line disables the header of the file.
- ⑦ This line indicates to the system to load the existing data from the archive on start up. Without this, only new realtime data are exported.

- ⑧ This line hides the menu entry for the component. Since this component accepts no commands and produces no data, this is often desirable.

Full System

This example is of a full acquisition system and combines many of the above components into a final complete system.

```
/aerosol/Autoprobe/DisableDefaultComponents,TRUE
/aerosol/Components/BypassFailsafe/Actions/#0/Bypass,Bypass|ImpactorPressureTooHigh
/aerosol/Components/BypassFailsafe/Actions/#0/Continuous,TRUE
/aerosol/Components/BypassFailsafe/Actions/#0/Dependencies/Acquire,|ImpactorPressureBy
pass
/aerosol/Components/BypassFailsafe/Actions/#0/Dependencies/Hold,30.0
/aerosol/Components/BypassFailsafe/Actions/#0/Event,"Pressure across the impactor is
too high. System bypassed. Please check the manual ball valve."
/aerosol/Components/BypassFailsafe/Actions/#0/EventRealtime,TRUE
/aerosol/Components/BypassFailsafe/Actions/#0/Triggers/#0/Input,"rt_instant:Pd_P11"
/aerosol/Components/BypassFailsafe/Actions/#0/Triggers/#0/Threshold,100.0
/aerosol/Components/BypassFailsafe/Actions/#0/Triggers/#0/Type,"Threshold"
/aerosol/Components/BypassFailsafe/Actions/#1/ClearBypass,|ImpactorPressureTooHigh
/aerosol/Components/BypassFailsafe/Actions/#1/Dependencies/Release,|ImpactorPressureBy
pass
/aerosol/Components/BypassFailsafe/Actions/#1/Event,"Pressure across the impactor
returned to normal. The system lock has been released but a manual release is
required before sampling resumes."
/aerosol/Components/BypassFailsafe/Actions/#1/EventRealtime,TRUE
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Input,"rt_instant:Pd_P11"
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Operation,"Less"
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Require,|ImpactorPressureBy
pass
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Threshold,90.0
/aerosol/Components/BypassFailsafe/Actions/#1/Triggers/#0/Type,"Threshold"
/aerosol/Components/BypassFailsafe/Fixed,TRUE
/aerosol/Components/BypassFailsafe/Groups,control|sample
/aerosol/Components/BypassFailsafe/MenuCharacter,""
/aerosol/Components/BypassFailsafe/MenuEntry,"Bypass Failsafe"
/aerosol/Components/BypassFailsafe/Name,"control_trigger"
/aerosol/Components/CLAP/Area/#0,19.86
/aerosol/Components/CLAP/Area/#1,19.75
/aerosol/Components/CLAP/Area/#2,19.86
/aerosol/Components/CLAP/Area/#3,19.88
/aerosol/Components/CLAP/Area/#4,20.05
/aerosol/Components/CLAP/Area/#5,20.27
/aerosol/Components/CLAP/Area/#6,20.45
/aerosol/Components/CLAP/Area/#7,20.36
/aerosol/Components/CLAP/FlowScale,0.99
/aerosol/Components/CLAP/Groups,sample|spancheck
/aerosol/Components/CLAP/HardwareFlowCalibration/#0,-2.3739
/aerosol/Components/CLAP/HardwareFlowCalibration/#1,0.9981
```

```

/aerosol/Components/CLAP/HardwareFlowCalibration/#2,0.0
/aerosol/Components/CLAP/HardwareFlowCalibration/#3,0.0
/aerosol/Components/CLAP/Instrument,"A11"
/aerosol/Components/CLAP/Match/SerialNumber,19
/aerosol/Components/CLAP/Name,"acquire_gmd_clap3w"
/aerosol/Components/CLAP/VariableMetadata/Q_.*/*dNoteCalibration,"10.019#2016-02-19/EJA@THD;10.019#2012-02-28/PJS@GMD"
/aerosol/Components/CLAP/VariableMetadata/Q_.*/*dNoteScale,"10.019#2012-02-28/PJS@GMD"
/aerosol/Components/CPC/Flow,1.346
/aerosol/Components/CPC/Groups,|cpc
/aerosol/Components/CPC/Instrument,"N71"
/aerosol/Components/CPC/MenuCharacter,"C"
/aerosol/Components/CPC/Name,"acquire_gmd_cpcpulse"
/aerosol/Components/CPC/Source/Manufacturer,"TSI"
/aerosol/Components/CPC/Source/Model,"3760"
/aerosol/Components/CPC/Source/SerialNumber,405
/aerosol/Components/CPC/VariableMetadata/N_.*/*dNoteFlow,"1.346 (22.43) 20160222/eja
1.47 (24.5) 20150111/eja; 24.5 #131211/eja; 24.3517 #121208/eja;22.1667 #111214/eja;
24.9 #091510eja #24.0; 091016/PJS@THD"
/aerosol/Components/CPCDrierFlow/Groups,|cpc
/aerosol/Components/CPCDrierFlow/Instrument,"Q72"
/aerosol/Components/CPCDrierFlow/Match/SerialNumber,412110911001
/aerosol/Components/CPCDrierFlow/MenuCharacter,"F"
/aerosol/Components/CPCDrierFlow/MenuEntry,"CPC Drier Flow"
/aerosol/Components/CPCDrierFlow/Name,"acquire_tsi_mfm4xxx"
/aerosol/Components/CPCDrierFlow/Remap/Q/Calibration/#0,-0.2749
/aerosol/Components/CPCDrierFlow/Remap/Q/Calibration/#1,1.2845
/aerosol/Components/CPCDrierFlow/VariableMetadata/Q_.*/*dDescription,"CPC drier flow"
/aerosol/Components/CPCDrierFlow/VariableMetadata/Q_.*/*dNoteCalibration,"-
0.2749;1.2845 160219/eja-0.013;1.093 150111/eja; #0;1.02 #131211/eja; -0.0401;1.0013
#121208/eja;-0.058,0.969 #111214/eja; 0.454;1.08 #100919/eja;-
0.05,1.02;091001/EJA@GMD-this is SLPM"
/aerosol/Components/CPCSampleFlow/Groups,|cpc
/aerosol/Components/CPCSampleFlow/Instrument,"Q71"
/aerosol/Components/CPCSampleFlow/Match/SerialNumber,412110912002
/aerosol/Components/CPCSampleFlow/MenuCharacter,"E"
/aerosol/Components/CPCSampleFlow/MenuEntry,"CPC Sample Flow"
/aerosol/Components/CPCSampleFlow/Name,"acquire_tsi_mfm4xxx"
/aerosol/Components/CPCSampleFlow/Remap/Q/Calibration/#0,-0.0113
/aerosol/Components/CPCSampleFlow/Remap/Q/Calibration/#1,1.0233
/aerosol/Components/CPCSampleFlow/VariableMetadata/Q_.*/*dDescription,"CPC sample
flow"
/aerosol/Components/CPCSampleFlow/VariableMetadata/Q_.*/*dNoteCalibration,"-
0.0113;1.0233 20160219/eja;-0.007;0.822 20150111/eja; #-0.007;1.02 #131211/eja;
-0.0231;1.0022 #121208/eja;-0.007,1.036 #111214/eja; 0.06;0.98 #back to old cal, new
cal prob messed by tube crack;-0.038,0.545;100919/eja;0.06,0.98;091001/EJA@GMD-this is
SLPM"
/aerosol/Components/ImpactorSchedule/Actions/#0/Aligned,TRUE
/aerosol/Components/ImpactorSchedule/Actions/#0/Count,0
/aerosol/Components/ImpactorSchedule/Actions/#0/InstrumentCommands/X1/SetDigital/PM1Im
pactor,TRUE

```



```
/aerosol/Components/ImpactorSchedule/Actions/#0/SetCut,"pm1"  
/aerosol/Components/ImpactorSchedule/Actions/#0/Unit,"minute"  
/aerosol/Components/ImpactorSchedule/Actions/#1/Aligned,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#1/Count,6  
/aerosol/Components/ImpactorSchedule/Actions/#1/InstrumentCommands/X1/SetDigital/PM1Im  
pactor,FALSE  
/aerosol/Components/ImpactorSchedule/Actions/#1/SetCut,"pm10"  
/aerosol/Components/ImpactorSchedule/Actions/#1/Unit,"minute"  
/aerosol/Components/ImpactorSchedule/Actions/#2/Aligned,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#2/Count,12  
/aerosol/Components/ImpactorSchedule/Actions/#2/InstrumentCommands/X1/SetDigital/PM1Im  
pactor,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#2/SetCut,"pm1"  
/aerosol/Components/ImpactorSchedule/Actions/#2/Unit,"minute"  
/aerosol/Components/ImpactorSchedule/Actions/#3/Aligned,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#3/Count,18  
/aerosol/Components/ImpactorSchedule/Actions/#3/InstrumentCommands/X1/SetDigital/PM1Im  
pactor,FALSE  
/aerosol/Components/ImpactorSchedule/Actions/#3/SetCut,"pm10"  
/aerosol/Components/ImpactorSchedule/Actions/#3/Unit,"minute"  
/aerosol/Components/ImpactorSchedule/Actions/#4/Aligned,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#4/Count,24  
/aerosol/Components/ImpactorSchedule/Actions/#4/InstrumentCommands/X1/SetDigital/PM1Im  
pactor,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#4/SetCut,"pm1"  
/aerosol/Components/ImpactorSchedule/Actions/#4/Unit,"minute"  
/aerosol/Components/ImpactorSchedule/Actions/#5/Aligned,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#5/Count,30  
/aerosol/Components/ImpactorSchedule/Actions/#5/InstrumentCommands/X1/SetDigital/PM1Im  
pactor,FALSE  
/aerosol/Components/ImpactorSchedule/Actions/#5/SetCut,"pm10"  
/aerosol/Components/ImpactorSchedule/Actions/#5/Unit,"minute"  
/aerosol/Components/ImpactorSchedule/Actions/#6/Aligned,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#6/Count,36  
/aerosol/Components/ImpactorSchedule/Actions/#6/InstrumentCommands/X1/SetDigital/PM1Im  
pactor,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#6/SetCut,"pm1"  
/aerosol/Components/ImpactorSchedule/Actions/#6/Unit,"minute"  
/aerosol/Components/ImpactorSchedule/Actions/#7/Aligned,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#7/Count,42  
/aerosol/Components/ImpactorSchedule/Actions/#7/InstrumentCommands/X1/SetDigital/PM1Im  
pactor,FALSE  
/aerosol/Components/ImpactorSchedule/Actions/#7/SetCut,"pm10"  
/aerosol/Components/ImpactorSchedule/Actions/#7/Unit,"minute"  
/aerosol/Components/ImpactorSchedule/Actions/#8/Aligned,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#8/Count,48  
/aerosol/Components/ImpactorSchedule/Actions/#8/InstrumentCommands/X1/SetDigital/PM1Im  
pactor,TRUE  
/aerosol/Components/ImpactorSchedule/Actions/#8/SetCut,"pm1"  
/aerosol/Components/ImpactorSchedule/Actions/#8/Unit,"minute"  
/aerosol/Components/ImpactorSchedule/Actions/#9/Aligned,TRUE
```

```

/aerosol/Components/ImpactorSchedule/Actions/#9/Count,54
/aerosol/Components/ImpactorSchedule/Actions/#9/InstrumentCommands/X1/SetDigital/PM1Impactor, FALSE
/aerosol/Components/ImpactorSchedule/Actions/#9/SetCut,"pm10"
/aerosol/Components/ImpactorSchedule/Actions/#9/Unit,"minute"
/aerosol/Components/ImpactorSchedule/DefaultFlushTime,62.0
/aerosol/Components/ImpactorSchedule/Fixed,TRUE
/aerosol/Components/ImpactorSchedule/Groups,|sample
/aerosol/Components/ImpactorSchedule/Instrument,"IMPACTOR"
/aerosol/Components/ImpactorSchedule/MenuCharacter,""
/aerosol/Components/ImpactorSchedule/MenuEntry,"Impactor Schedule"
/aerosol/Components/ImpactorSchedule/Name,"control_cycle"
/aerosol/Components/MCPC/Groups,|cpc
/aerosol/Components/MCPC/Instrument,"N72"
/aerosol/Components/MCPC/MenuCharacter,"D"
/aerosol/Components/MCPC/Name,"acquire_bmi_cpc1720"
/aerosol/Components/Neph/Groups,sample|spancheck
/aerosol/Components/Neph/Name,"acquire_tsi_neph3563"
/aerosol/Components/Neph/Source/SerialNumber,1083
/aerosol/Components/Ozone/Address,49
/aerosol/Components/Ozone/GlobalMetadata/*dCalibration/Date,"20110611"
/aerosol/Components/Ozone/GlobalMetadata/*dCalibration/Outcome,"pass"
/aerosol/Components/Ozone/GlobalMetadata/*dCalibration/ReferenceDate,"20070630"
/aerosol/Components/Ozone/GlobalMetadata/*dCalibration/Uncertainty,2.0
/aerosol/Components/Ozone/Groups,|ozone
/aerosol/Components/Ozone/Instrument,"G81"
/aerosol/Components/Ozone/MenuCharacter,"G"
/aerosol/Components/Ozone/Name,"acquire_thermo_ozone49"
/aerosol/Components/Ozone/Source/SerialNumber,323000000000552
/aerosol/Components/PID/Controllers/#0/Address,49
/aerosol/Components/PID/Controllers/#0/Setpoint,40.0
/aerosol/Components/PID/Controllers/#1/Address,50
/aerosol/Components/PID/Controllers/#1/Setpoint,40.0
/aerosol/Components/PID/Controllers/#2/Address,51
/aerosol/Components/PID/Controllers/#2/Setpoint,55.0
/aerosol/Components/PID/Controllers/#3/Address,52
/aerosol/Components/PID/Controllers/#3/ManualMode,TRUE
/aerosol/Components/PID/Controllers/#3/Setpoint,29.5
/aerosol/Components/PID/Groups,|control
/aerosol/Components/PID/Instrument,"X2"
/aerosol/Components/PID/MenuEntry,"Love PID"
/aerosol/Components/PID/Name,"acquire_love_pid"
/aerosol/Components/PID/VariableGroups/#0/Groups,|sample
/aerosol/Components/PID/VariableGroups/#0/Variable,"U_V11"
/aerosol/Components/PID/VariableGroups/#1/Groups,|sample
/aerosol/Components/PID/VariableGroups/#1/Variable,"Q_Q11"
/aerosol/Components/PID/VariableGroups/#2/Groups,|control
/aerosol/Components/PID/VariableGroups/#2/Variable,".*_X2"
/aerosol/Components/PID/Variables/Q_Q11/Address,52
/aerosol/Components/PID/Variables/Q_Q11/Calibration/#0,1.75
/aerosol/Components/PID/Variables/Q_Q11/Calibration/#1,0.897

```

```

/aerosol/Components/PID/Variables/Q_Q11/Metadata/*dDescription,"Analyzer flow"
/aerosol/Components/PID/Variables/Q_Q11/Metadata/*dNoteCalibration,"1.75;0.897
20160219/eja; 1.54;0.89 20150110/eja; 2.02;0.88 #131211/eja; 1.9127;0.8871
#121208/eja;0.861,0.912 #111214/eja; 0.278;0.974 #100919/eja;0.164,1.00;091001/eja"
/aerosol/Components/PID/Variables/Q_Q11/Metadata/*dUnits,"lpm"
/aerosol/Components/PID/Variables/U_V02/Address,50
/aerosol/Components/PID/Variables/U_V02/Calibration/#0,0.0
/aerosol/Components/PID/Variables/U_V02/Calibration/#1,1.0
/aerosol/Components/PID/Variables/U_V02/Metadata/*dDescription,"Rack RH"
/aerosol/Components/PID/Variables/U_V02/Metadata/*dUnits,"%"
/aerosol/Components/PID/Variables/U_V11/Address,51
/aerosol/Components/PID/Variables/U_V11/Calibration/#0,0.0
/aerosol/Components/PID/Variables/U_V11/Calibration/#1,1.0
/aerosol/Components/PID/Variables/U_V11/Metadata/*dDescription,"Impactor box inlet RH"
/aerosol/Components/PID/Variables/U_V11/Metadata/*dNoteCalibration,"0.07;0.92
#131211/eja; 0;1 #130117/eja; -3.1;1.0167 # 091016/PJS@THDsame as inlet RH"
/aerosol/Components/PID/Variables/U_V11/Metadata/*dUnits,"%"
/aerosol/Components/PID/Variables/U_V51/Address,49
/aerosol/Components/PID/Variables/U_V51/Calibration/#0,0.0
/aerosol/Components/PID/Variables/U_V51/Calibration/#1,1.0
/aerosol/Components/PID/Variables/U_V51/Metadata/*dDescription,"Splitter RH"
/aerosol/Components/PID/Variables/U_V51/Metadata/*dNoteCalibration,"0.0;1.0
#20150110/eja 0.79;0.97 #131211/eja; 0;1 #130117/eja;-1.778;1.0937 #090212/PJS"
/aerosol/Components/PID/Variables/U_V51/Metadata/*dUnits,"%"
/aerosol/Components/ZeroSchedule/Actions/#0/Aligned,TRUE
/aerosol/Components/ZeroSchedule/Actions/#0/Commands/StartZero,TRUE
/aerosol/Components/ZeroSchedule/Actions/#0/Count,3418
/aerosol/Components/ZeroSchedule/Actions/#0/Description,"System Zero"
/aerosol/Components/ZeroSchedule/Actions/#0/Unit,"second"
/aerosol/Components/ZeroSchedule/Fixed,TRUE
/aerosol/Components/ZeroSchedule/Groups,|sample
/aerosol/Components/ZeroSchedule/MenuCharacter,""
/aerosol/Components/ZeroSchedule/MenuEntry,"Zero Schedule"
/aerosol/Components/ZeroSchedule/Name,"control_cycle"
/aerosol/Components/uMAC/Bypass/Digital/DisableBypassValve,FALSE
/aerosol/Components/uMAC/DigitalOutputs/DisableBypassValve,0
/aerosol/Components/uMAC/DigitalOutputs/PM1Impactor,1
/aerosol/Components/uMAC/Exit/Digital/DisableBypassValve,FALSE
/aerosol/Components/uMAC/Groups,control|spancheck
/aerosol/Components/uMAC/Initialize/Digital/DisableBypassValve,TRUE
/aerosol/Components/uMAC/Instrument,"X1"
/aerosol/Components/uMAC/Name,"acquire_azonix_umac1050"
/aerosol/Components/uMAC/UnBypass/Digital/DisableBypassValve,TRUE
/aerosol/Components/uMAC/VariableGroups/#0/Groups,|sample
/aerosol/Components/uMAC/VariableGroups/#0/Variable,"T_V11"
/aerosol/Components/uMAC/VariableGroups/#1/Groups,|sample
/aerosol/Components/uMAC/VariableGroups/#1/Variable,"Pd_P11"
/aerosol/Components/uMAC/VariableGroups/#2/Groups,|sample
/aerosol/Components/uMAC/VariableGroups/#2/Variable,"Pd_P12"
/aerosol/Components/uMAC/VariableGroups/#3/Groups,|control
/aerosol/Components/uMAC/VariableGroups/#3/Variable,".*_X1"

```

```

/aerosol/Components/uMAC/VariableGroups/#4/Groups,|control
/aerosol/Components/uMAC/VariableGroups/#4/Variable,"(T|(Pd?)|U|Q)_(T|P|V|Q)[560][0-9]"
/aerosol/Components/uMAC/Variables/Pd_P01/Calibration/#0,-0.504
/aerosol/Components/uMAC/Variables/Pd_P01/Calibration/#1,4.523
/aerosol/Components/uMAC/Variables/Pd_P01/Channel,19
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dDescription,"Stack pitot tube"
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dFormat,"00.000"
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dGroupUnits,"dhPa"
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dNoteCalibration,"-0.504;4.523
20160220/eja; -0.6139;4.5888 20150110/eja; 0.0029;4.33 #131211/eja; -0.2181;4.2041
#121208/eja;-0.54,4.11 #111214/eja; -0.78;4.189 # 100919/eja;-0.806,4.283;100407/MI;-
0.25,4.03;080418/eja"
/aerosol/Components/uMAC/Variables/Pd_P01/Metadata/*dUnits,"hPa"
/aerosol/Components/uMAC/Variables/Pd_P11/Calibration/#0,-186.3
/aerosol/Components/uMAC/Variables/Pd_P11/Calibration/#1,333.75
/aerosol/Components/uMAC/Variables/Pd_P11/Channel,20
/aerosol/Components/uMAC/Variables/Pd_P11/Metadata/*dDescription,"Impactor pressure
drop"
/aerosol/Components/uMAC/Variables/Pd_P11/Metadata/*dGroupUnits,"dhPa"
/aerosol/Components/uMAC/Variables/Pd_P11/Metadata/*dNoteCalibration,"-
186.3;333.75#20170320/eja@GMD; -197.97;139.34#20160219/eja;
-194.79;137.3#20150110/eja; -194.03;135.99 #131211/eja; -192.04;135.34 #121208/eja-
193.96,136.48 #111214/eja; -198.45;139.82 #100919/eja;-205.67,144.10;090930/eja; calib
w/0-15 in h2o magnehelic"
/aerosol/Components/uMAC/Variables/Pd_P11/Metadata/*dUnits,"hPa"
/aerosol/Components/uMAC/Variables/Pd_P12/Calibration/#0,-186
/aerosol/Components/uMAC/Variables/Pd_P12/Calibration/#1,333.18
/aerosol/Components/uMAC/Variables/Pd_P12/Channel,21
/aerosol/Components/uMAC/Variables/Pd_P12/Metadata/*dDescription,"Pump vacuum"
/aerosol/Components/uMAC/Variables/Pd_P12/Metadata/*dFormat,"0000.00"
/aerosol/Components/uMAC/Variables/Pd_P12/Metadata/*dGroupUnits,"dhPa"
/aerosol/Components/uMAC/Variables/Pd_P12/Metadata/*dNoteCalibration,"-186.00; 333.18
20170320/eja@GMD; -196.72;145.88 20160219/eja; -211.16;144.46 20150110/eja;
-209.77;143.50 #131211/eja; -207.24;146.22 #121208/eja-203.07,145.37 #111214/eja;
-208.7;143.95 #100919/eja-317.12,212.64;090930/eja; calib w/0-10 psi magnehelic"
/aerosol/Components/uMAC/Variables/Pd_P12/Metadata/*dUnits,"hPa"
/aerosol/Components/uMAC/Variables/T_V02/Calibration/#0,-40
/aerosol/Components/uMAC/Variables/T_V02/Calibration/#1,100.0
/aerosol/Components/uMAC/Variables/T_V02/Channel,8
/aerosol/Components/uMAC/Variables/T_V02/Metadata/*dDescription,"Rack temperature"
/aerosol/Components/uMAC/Variables/T_V02/Metadata/*dUnits,"°C"
/aerosol/Components/uMAC/Variables/T_V11/Calibration/#0,-40
/aerosol/Components/uMAC/Variables/T_V11/Calibration/#1,100.0
/aerosol/Components/uMAC/Variables/T_V11/Channel,9
/aerosol/Components/uMAC/Variables/T_V11/Metadata/*dDescription,"Impactor box inlet
temperature"
/aerosol/Components/uMAC/Variables/T_V11/Metadata/*dNoteCalibration,"20150111/eja; #-
28.0;100.0 #130117/eja"
/aerosol/Components/uMAC/Variables/T_V11/Metadata/*dUnits,"°C"
/aerosol/Components/uMAC/Variables/T_V51/Calibration/#0,-40

```

```
/aerosol/Components/uMAC/Variables/T_V51/Calibration/#1,100.0  
/aerosol/Components/uMAC/Variables/T_V51/Channel,10  
/aerosol/Components/uMAC/Variables/T_V51/Metadata/*dDescription,"Splitter temperature"  
/aerosol/Components/uMAC/Variables/T_V51/Metadata/*dNoteCalibration,"20150110/eja  
-29.0;100.0 #130117/eja"  
/aerosol/Components/uMAC/Variables/T_V51/Metadata/*dUnits,"°C"
```

acquisition variable

The `acquisition` variable in the `CONFIGURATION` archive controls the realtime acquisition system.

In general this configuration is split by profiles to allow for multiple acquisition configuration on the same station. However, this is normally only useful concurrently if there are multiple systems accessing the same storage database from multiple computers. To run multiple acquisition systems on the same computer, various lockouts must be disabled and is highly recommended against. Instead, simply create multiple groups of acquisition components to designate the various "systems" running from the same computer.

This section contains the full list of possibilities for the configuration. In general, this is mostly a reference to be used with existing configurations. For creating one from scratch, using the graphical editor or starting from an existing one is highly recommended.

General Settings

/

Description

Acquisition system configuration

Keys

The keys immediately below this are profile names that are selected by the application. For example, `aerosol` is the commonly used default profile. However, any non-empty name is acceptable and can be used to create customized configurations that can be selected at run time. In general, profile names should be all lower case to preserve case insensitivity. In the paths listed `<PROFILE>` is used as a placeholder to real profile name (e.g. replace `<PROFILE>` with `aerosol` for the default profile).

`/<PROFILE>/AllowMultiple`

Description

Allow multiple instances of the acquisition system to run at the same time

Type

`Boolean`, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Archive/Immediate`

Description

Immediately write all data to disk instead of queing it up in batches

Type

`Boolean`, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Archive/PurgeAge`

Description

The maximum number of seconds before the current time to allow suffix based data removal

Type

Real number

Undefined values

Not permitted

`<PROFILE>/Archive/PurgePreserveDuration`**Description**

The minimum number of seconds in duration to preserve existing values during suffix based removal

Type

Real number

Undefined values

Not permitted

`<PROFILE>/Autoprobe/Additional`**Description**

Additional interfaces to attempt auto-detection on beyond any system default ones added

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`<PROFILE>/Autoprobe/Additional/#0`**Type**

I/O Interface

Structure

An I/O interface is a general purpose endpoint used to communicate with an external system. The simplest case is a local serial port connected to an instrument. The possible valid configuration parameters are determined by the `Type enumeration` key.

`<PROFILE>/Autoprobe/Blacklist`**Description**

A list of system default ports to exclude from auto-detection

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`<PROFILE>/Autoprobe/Blacklist/#0`**Type**

I/O Interface

Structure

An I/O interface is a general purpose endpoint used to communicate with an external system. The simplest case is a local serial port connected to an instrument. The possible valid configuration parameters are determined by the `Type enumeration` key.

`<PROFILE>/Autoprobe/DefaultFixedInterface`

Description

Do not allow auto-detected interfaces to change interface parameters by default

Type

`Boolean`, values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Autoprobe/DefaultIgnoreBackgroundFailure`

Description

Ignore failures of background detection by default

Type

`Boolean`, values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Autoprobe/DefaultPassive`

Description

Do not allow auto-detected components to write to the interfaces by default

Type

`Boolean`, values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Autoprobe/Disable`

Description

Disable all auto-detection

Type

`Boolean`, values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Autoprobe/DisableDefaultComponents`

Description

Disable default configured components and use only explicitly defined ones

Type

`Boolean`, values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Autoprobe/DisableDefaultInterfaces`

Description

Disable all auto-detected system ports

Type

`Boolean`, values are `TRUE` or `FALSE` and defaults to `FALSE`

/<PROFILE>/Autoprobe/FailsafeTimeout

Description

The maximum time to set as a timeout after interface promotion from auto-detection

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

/<PROFILE>/Autoprobe/Retries

Description

The number of times to try auto-detection again for all possibilities after a failure

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

/<PROFILE>/Autoprobe/RetryOnEmpty

Description

Retry auto-detection if no components succeed

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/AveragingInterval

Description

The averaging interval

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting. Alignment is enabled by default.

Undefined values

One minute aligned

`<PROFILE>/AveragingInterval/Units`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`<PROFILE>/AveragingInterval/Count`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/AveragingInterval/Align`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/ComponentGroups`

Description

Group definitions based on matching component types and names

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices.

/<PROFILE>/ComponentGroups/#0/Component

Description

The acquisition component name to match as a regular expression

Type

Text

Undefined values

Any

/<PROFILE>/ComponentGroups/#0/Instrument

Description

The instrument suffix to match as a regular expression

Type

Text

Undefined values

Any

/<PROFILE>/Components

Description

The explicitly defined acquisition components

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed <IDENTIFIER> is used as a placeholder to real name selected during configuration.

/<PROFILE>/Components/<IDENTIFIER>/Fixed

Description

Always define the component without involving auto-detection

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/GlobalMetadata

Description

Metadata to overlay for all instrument generated data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Groups`

Description

The groups the component belongs to for input state

Type

[Flags](#), a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/IgnoreBackgroundFailure`

Description

Ignore failures of background detection

Type

[Boolean](#), values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/InitialOrder`

Description

The initial ordering of components within a priority for autodetection

Type

[Integer](#)

Undefined values

Automatic

`/<PROFILE>/Components/<IDENTIFIER>/Instrument`

Description

The instrument suffix in use

Type

[Text](#)

Undefined values

Automatically Constructed

`/<PROFILE>/Components/<IDENTIFIER>/Interface`

Description

The backend interface used by the component for communications

Type

[I/O Interface](#)

Structure

An I/O interface is a general purpose endpoint used to communicate with an external system. The simplest case is a local serial port connected to an instrument. The possible valid

configuration parameters are determined by the [Type enumeration](#) key.

`<PROFILE>/Components/<IDENTIFIER>/Line`

Description

The sample line number to use in instrument identifier construction

Type

[Integer](#)

Valid range

$1 \leq \text{value}$

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/MenuCharacter`

Description

The shortcut menu character used

Type

[Text](#)

Undefined values

Automatically selected

`<PROFILE>/Components/<IDENTIFIER>/MenuEntry`

Description

The text disabled in the main menu list

Type

[Text](#)

Undefined values

Automatically generated

`<PROFILE>/Components/<IDENTIFIER>/MenuHide`

Description

Do not list the component in the main menu

Type

[Boolean](#), values are [TRUE](#) or [FALSE](#) and defaults to [FALSE](#)

`<PROFILE>/Components/<IDENTIFIER>/Name`

Description

The acquisition component name in use

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/OutputGroups`

Description

The groups the component generates control state for

Type

Flags, a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/Passive`

Description

Disable all output to the control interface and only read from it

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Priority`

Description

The sort priority for component creation and detection ordering

Type

Integer

Undefined values

Automatic

`/<PROFILE>/Components/<IDENTIFIER>/Remap`

Description

The remappings applied to raw variable identifiers during initial conversion

Keys

The keys immediately below this are internal variable names. These variables are the internal identifiers used by the acquisition component to control its acquisition. In most cases the final output values correspond to just their name without the suffix. For example BaG_A11 might have the internal identifier BaG. In the paths listed <VARIABLE> is used as a placeholder to real value used during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Remap/<VARIABLE>/Calibration`

Description

A calibration applied to the input value

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with real number values. For example, #0 is the constant

offset while #1 is the slope.

`</PROFILE>/Components/<IDENTIFIER>/Remap/<VARIABLE>/Input`

Description

An alternate realtime value to replace the value with

Type

Sequence Selection

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

`</PROFILE>/Components/<IDENTIFIER>/Retries`

Description

The number of times to try auto-detection again for all possibilities after a failure

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Source`

Description

The override source parameters for data generated by the component

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`</PROFILE>/Components/<IDENTIFIER>/Source/Manufacturer`

Description

The instrument manufacturer

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Source/Model`

Description

The instrument model identifier

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Source/SerialNumber`

Description

The instrument serial number

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Station`

Description

The station to assign the component to

Type

Text

Undefined values

Main acquisition station

`/<PROFILE>/Components/<IDENTIFIER>/VariableGroups`

Description

The group definitions applied to output variables made by the interface

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/VariableGroups/#0/Groups`

Description

The groups the selected data belongs to for state control

Type

Flags, a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/VariableGroups/#0/Variable`

Description

The matched variable the grouping applies to, this can also be a full sequence selection for complex configurations

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/VariableMetadata`

Description

Metadata to overlay for specific variables matched from those component generates

Keys

The keys immediately below this are regular expressions matched against the input. These regular expressions are non-empty text that constructs a pattern matcher. In the simplest form, this is just the full text, but more complex expressions are supported. In the paths listed `<REGEXP>` is used as a placeholder to real value used during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/VariableMetadata/<REGEXP>`

Description

Specific variable metadata

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/WindowTitle`

Description

The text used as a title for the realtime display window

Type

Text

Undefined values

The menu entry

`/<PROFILE>/Components/<IDENTIFIER>/FixedInterface`

Excluding conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Fixed = TRUE`

Description

Disable any alteration of the control interface parameters (e.x. baud)

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Match`

Excluding conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Fixed = TRUE`

Description

The required matching parameters for the interface to pass autodetection

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Match/Manufacturer`

Excluding conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Fixed = TRUE`

Description

The instrument reported manufacturer

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Match/Model`**Excluding conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Fixed = TRUE`

Description

The instrument reported model identifier

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Match/SerialNumber`**Excluding conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Fixed = TRUE`

Description

The instrument reported serial number to require

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/DataUpdateInterval`**Description**

The numbers of second between variable group data updates

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

`/<PROFILE>/DisableFileLockout`

Description

Disable file lock out checking on supported operating systems

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/DisableLocal`

Description

Disable the server local socket

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/DisableLoopBack`

Description

Disable the server loopback TCP socket

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/EnableRealtimeDiscardInstant`

Description

The default state for realtime instantaneous data discarding (e.x. disabling data during bypassing)

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/EnableRealtimeStatistics`

Description

Enable generation of value statistics for the realtime data stream

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/EnableStatistics`

Description

Enable generation of value statistics

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Listen`

Description

A list of network addresses to listen for client connections on

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices.

`/<PROFILE>/Listen/#0/Address`

Description

The address to listen on

Type

Text

Undefined values

Any

`/<PROFILE>/Listen/#0/Port`

Description

The port to listen on

Type

Integer

Valid range

$1 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Listen/#0/SSL`

Description

SSL parameters used for connection

Type

SSL Specification

Structure

A SSL specification consists of a hash map defining the parameters for encrypted SSL/TLS connections. The specification also permits enforcing peer certificate validity.

`/<PROFILE>/Listen/#0/SSL/Authorization`

Description

SSL certificate white list

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices.

`<PROFILE>/Listen/#0/SSL/Authorization/#0`

Description

SSL certificate

Type

Text

`<PROFILE>/LocalSocket`

Description

Local socket name that clients can connect to

Type

Text

Undefined values

Any

`<PROFILE>/LoopbackPort`

Description

Loopback TCP port that clients can connect to

Type

Integer

Valid range

$1 \leq \text{value}$

Undefined values

Not permitted

`<PROFILE>/State`

Description

The prefix applied to the internal state keys

Type

Text

`<PROFILE>/VariableGroups`

Description

The global group definitions applied to all variables

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`<PROFILE>/VariableGroups/#0/Groups`

Description

The groups the selected data belongs to for state control

Type

Flags, a set of unique text strings

`</PROFILE>/VariableGroups/#0/Variable`

Description

The matched variable the grouping applies to, this can also be a full sequence selection for complex configurations

Type

Text

2B Ozone Monitor Model 205

`</PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_2b_ozone205`

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_2b_ozone205`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Aerosol Dynamic Magic CPC Model 200

/<PROFILE>/Components/<IDENTIFIER>/CalculateConcentration

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ad_cpcmagic200

Description

Recalculate the concentration from the reported count rate

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Flow

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ad_cpcmagic200

Description

Counter sample flow rate

Type

Real number

Valid range

0.0 < value

Units

lpm

Undefined values

Instrument reported

/<PROFILE>/Components/<IDENTIFIER>/ReportInterval

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ad_cpcmagic200

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ad_cpcmagic200`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/UseMeasuredTime`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ad_cpcmagic200`

Description

Use the measured time between records for calculation instead of the instrument reported sample time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Aerodyne CAPS

`<PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_aerodyne_caps`

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_aerodyne_caps`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

Azonix μ MAC

`/<PROFILE>/Components/<IDENTIFIER>/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Instrument address

Type

Integer

Valid range

$0 \leq \text{value} \leq 63$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/AnalogChannelCount`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Total number of analog input channels

Type

Integer

Valid range

$1 \leq \text{value} \leq 62$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/AnalogChannelStart`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Start index of analog input channels

Type

Integer

Valid range

$0 \leq \text{value} \leq 61$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/AnalogOutputs**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050

Description

Analog output name to channel number mappings

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed <IDENTIFIER> is used as a placeholder to real name selected during configuration.

/<PROFILE>/Components/<IDENTIFIER>/AnalogOutputs/<IDENTIFIER>**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050

Description

Analog output channel number

Type

Integer

Valid range

$0 \leq \text{value} \leq 9$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Bypass/Analog**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050

Description

Analog output bypass values corresponding to defined analog output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Bypass/Analog/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Analog output value

Type

Real number

Valid range

$-10.0 \leq \text{value} \leq 10.0$

`/<PROFILE>/Components/<IDENTIFIER>/Bypass/Digital`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output bypass values corresponding to defined digital output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Bypass/Digital/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output state

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/DigitalOutputs`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output name to channel number mappings

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/DigitalOutputs/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output channel number

Type

Integer

Valid range

$0 \leq \text{value} \leq 61$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Analog`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Analog output shutdown values corresponding to defined analog output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Analog/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Analog output value

Type

Real number

Valid range

$-10.0 \leq \text{value} \leq 10.0$

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Digital`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output shutdown values corresponding to defined digital output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Digital/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output state

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Initialize/Analog`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Analog output initialization values corresponding to defined analog output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Initialize/Analog/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Analog output value

Type

Real number

Valid range

$-10.0 \leq \text{value} \leq 10.0$

`</PROFILE>/Components/<IDENTIFIER>/Initialize/Digital`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output initialization values corresponding to defined digital output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`</PROFILE>/Components/<IDENTIFIER>/Initialize/Digital/<IDENTIFIER>`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output state

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`</PROFILE>/Components/<IDENTIFIER>/LogOutputs`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Generate logged values based on the current output settings

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`</PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

1.0

`/<PROFILE>/Components/<IDENTIFIER>/UnBypass/Analog`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Analog output un-bypass values corresponding to defined analog output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/UnBypass/Analog/<IDENTIFIER>`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Analog output value

Type

Real number

Valid range

$-10.0 \leq \text{value} \leq 10.0$

`/<PROFILE>/Components/<IDENTIFIER>/UnBypass/Digital`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output un-bypass values corresponding to defined digital output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/UnBypass/Digital/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Digital output state

Type

[Boolean](#), values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Variables`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Defined output variables

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Calibration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Calibration applied to the raw input value

Type

[Calibration](#)

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with [real number](#) values. For example, `#0` is the constant offset while `#1` is the slope.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Channel`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Analog input channel number

Type

[Integer](#)

Valid range

$0 \leq \text{value} \leq 61$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050

Description

Base metadata for the variable

Structure

This value represents metadata injected directly into some output. That is, the exact contents defined by this value are used as metadata for another value. The exact keys are determined by what metadata is required.

/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dDescription**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050

Description

A sort description of the variable meaning

Type

Text

/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dGroupUnits**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050

Description

The units the variable is groups with for display

Type

Text

/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dNoteCalibration**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050

Description

Any notes about the calibration (location, date, etc)

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dUnits`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

The physical units the variable is reported in

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Name`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_azonix_umac1050`

Description

Output variable name

Type

Text

Undefined values

Key value

BMI CPC Model 1710

`/<PROFILE>/Components/<IDENTIFIER>/CalculateConcentration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_bmi_cpc1710`

Description

Recalculate the concentration from the reported count rate and flow

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_bmi_cpc1710`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

BMI CPC Model 1720

/<PROFILE>/Components/<IDENTIFIER>/CalculateConcentration**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_bmi_cpc1720

Description

Recalculate the concentration from the reported count rate and flow

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/PollInterval**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_bmi_cpc1720

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

Brooks PID Controller Model 0254

/<PROFILE>/Components/<IDENTIFIER>/Address**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254

Description

Instrument address or -1 for any

Type

Integer

Valid range

-1 ≤ value ≤ 65535

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Bypass/Setpoint`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Output bypass values corresponding to defined setpoint output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Bypass/Setpoint/<IDENTIFIER>`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Setpoint output value

Type

Real number

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Setpoint`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Output values set on shutdown corresponding to defined setpoint output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Setpoint/<IDENTIFIER>`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Setpoint output value

Type

Real number

`/<PROFILE>/Components/<IDENTIFIER>/Initialize/Setpoint`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Output initialization values corresponding to defined setpoint output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Initialize/Setpoint/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Setpoint output value

Type

Real number

`/<PROFILE>/Components/<IDENTIFIER>/LogOutputs`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Generate logged values based on the current output settings

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

1.0

/<PROFILE>/Components/<IDENTIFIER>/SetpointOutputs**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254

Description

Output name to controller setpoint number mappings

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed <IDENTIFIER> is used as a placeholder to real name selected during configuration.

/<PROFILE>/Components/<IDENTIFIER>/SetpointOutputs/<IDENTIFIER>**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254

Description

Setpoint channel number

Type

Integer

Valid range

$1 \leq \text{value} \leq 4$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/StrictMode**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

</PROFILE>/Components/<IDENTIFIER>/UnBypass/Setpoint

Required conditions

- **</PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254**

Description

Output un-bypass values corresponding to defined setpoint output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed **<IDENTIFIER>** is used as a placeholder to real name selected during configuration.

</PROFILE>/Components/<IDENTIFIER>/UnBypass/Setpoint/<IDENTIFIER>

Required conditions

- **</PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254**

Description

Setpoint output value

Type

Real number

</PROFILE>/Components/<IDENTIFIER>/Variables

Required conditions

- **</PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254**

Description

Defined output variables

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed **<IDENTIFIER>** is used as a placeholder to real name selected during configuration.

</PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Calibration

Required conditions

- **</PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254**

Description

Calibration applied to the raw input value

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with [real number](#) values. For example, `#0` is the constant offset while `#1` is the slope.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Channel`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Input channel number

Type

[Integer](#)

Valid range

$1 \leq \text{value} \leq 4$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

Base metadata for the variable

Structure

This value represents metadata injected directly into some output. That is, the exact contents defined by this value are used as metadata for another value. The exact keys are determined by what metadata is required.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dDescription`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254`

Description

A sort description of the variable meaning

Type

[Text](#)

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dGroupUnits`

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254

Description

The units the variable is groups with for display

Type

Text

/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dNoteCalibration

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254

Description

Any notes about the calibration (location, date, etc)

Type

Text

/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dUnits

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254

Description

The physical units the variable is reported in

Type

Text

/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Name

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_brooks_pid0254

Description

Output variable name

Type

Text

Undefined values

Key value

Campbell CR1000 with GMD firmware

/<PROFILE>/Components/<IDENTIFIER>/AnalogOutputs

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd

Description

Analog output name to channel number mappings

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/AnalogOutputs/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Analog output channel number

Type

Integer

Valid range

$0 \leq \text{value} \leq 7$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Bypass/Analog`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Analog output bypass values corresponding to defined analog output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Bypass/Analog/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Analog output value

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 5.0$

/<PROFILE>/Components/<IDENTIFIER>/Bypass/Digital**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd

Description

Digital output bypass values corresponding to defined digital output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed <IDENTIFIER> is used as a placeholder to real name selected during configuration.

/<PROFILE>/Components/<IDENTIFIER>/Bypass/Digital/<IDENTIFIER>**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd

Description

Digital output state

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/DigitalOutputs**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd

Description

Digital output name to channel number mappings

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed <IDENTIFIER> is used as a placeholder to real name selected during configuration.

/<PROFILE>/Components/<IDENTIFIER>/DigitalOutputs/<IDENTIFIER>**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd

Description

Digital output channel number

Type

Integer

Valid range

$0 \leq \text{value} \leq 20$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Analog`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Analog output shutdown values corresponding to defined analog output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Analog/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Analog output value

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 5.0$

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Digital`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Digital output shutdown values corresponding to defined digital output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Exit/Digital/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Digital output state

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Initialize/Analog`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Analog output initialization values corresponding to defined analog output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Initialize/Analog/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Analog output value

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 5.0$

`/<PROFILE>/Components/<IDENTIFIER>/Initialize/Digital`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Digital output initialization values corresponding to defined digital output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected

during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Initialize/Digital/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Digital output state

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/LogOutputs`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Generate logged values based on the current output settings

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

1.0

`/<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/UnBypass/Analog

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd

Description

Analog output un-bypass values corresponding to defined analog output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed <IDENTIFIER> is used as a placeholder to real name selected during configuration.

/<PROFILE>/Components/<IDENTIFIER>/UnBypass/Analog/<IDENTIFIER>

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd

Description

Analog output value

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 5.0$

/<PROFILE>/Components/<IDENTIFIER>/UnBypass/Digital

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd

Description

Digital output un-bypass values corresponding to defined digital output identifiers

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed <IDENTIFIER> is used as a placeholder to real name selected during configuration.

/<PROFILE>/Components/<IDENTIFIER>/UnBypass/Digital/<IDENTIFIER>

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd

Description

Digital output state

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Components/<IDENTIFIER>/Variables`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Defined output variables

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Calibration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Calibration applied to the raw input value

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with **real number** values. For example, `#0` is the constant offset while `#1` is the slope.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Channel`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Analog input channel number

Type

Integer

Valid range

$1 \leq \text{value} \leq 32$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Base metadata for the variable

Structure

This value represents metadata injected directly into some output. That is, the exact contents defined by this value are used as metadata for another value. The exact keys are determined by what metadata is required.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dDescription`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

A sort description of the variable meaning

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dGroupUnits`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

The units the variable is groups with for display

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dNoteCalibration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Any notes about the calibration (location, date, etc)

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dUnits`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

The physical units the variable is reported in

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Name`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_campbell_cr1000gmd`

Description

Output variable name

Type

Text

Undefined values

Key value

CSD POPS OPC

`/<PROFILE>/Components/<IDENTIFIER>/Diameter`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_csd_pops`

Description

The central diameters of the bins

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Diameter/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_csd_pops`

Description

Center bin diameter

Type

Real number

Valid range

0.0 < value

Units

µm

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_csd_pops`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/UseMeasuredTime`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_csd_pops`

Description

Use the measured time between records for count rate calculations instead of assuming 1 Hz

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

DMT BCP

`<PROFILE>/Components/<IDENTIFIER>/ConstantTime`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_bcp`

Description

Set the assumed constant time for count rate conversion

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Use measured time

`<PROFILE>/Components/<IDENTIFIER>/Diameter`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_bcp`

Description

The central diameters of the bins

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices. The array is expected to contain 10 values.

`/<PROFILE>/Components/<IDENTIFIER>/Diameter/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_bcp`

Description

Center bin diameter

Type

Real number

Valid range

0.0 < value

Units

µm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Flow`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_bcp`

Description

Flow through sample volume

Type

Real number

Valid range

0.0 < value

Units

lpm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Hardware`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_bcp`

Description

Hardware configuration parameters

Type

Real number

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_bcp`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

DMT CCN

`/<PROFILE>/Components/<IDENTIFIER>/BaseTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

Absolute time origin used when interpreting logged data

Type

Real number

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/BaseTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

Absolute time origin used when interpreting logged data

Structure

This is an absolute point in time specified as the number of seconds from the Unix epoch (1970-01-01T00:00:00Z).

`/<PROFILE>/Components/<IDENTIFIER>/Model/Enable`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

Enable model calculation of the supersaturation

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Model/Interval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

The length of time included in a single model run

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting. Alignment is enabled by default.

Undefined values

One minute aligned

`/<PROFILE>/Components/<IDENTIFIER>/Model/Interval/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Model/Interval/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Model/Interval/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Model/MaximumBacklog`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

The maximum number of model calculations outstanding before new ones are dropped

Type

Integer

Valid range

-1 ≤ value

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Model/Parameters`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

The contents of the parameters file used with the model program

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/Model/Program`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

The executable name of the model program

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/Processing/Interval`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

The length of time used for general processing (e.x. standard deviation calculations)

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting. Alignment is enabled by default.

Undefined values

One minute aligned

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Interval/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Interval/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Interval/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

Align units in logical time

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Components/<IDENTIFIER>/Processing/TemperatureStability`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

The smoother applied to the temperature difference used to determine if the data are stable

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the **Type enumeration** key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

The last valid point

`/<PROFILE>/Components/<IDENTIFIER>/Processing/UseCalculatedStability`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

Use the smoother calculated stability flag for logged data exclusion

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Components/<IDENTIFIER>/Processing/UseReportedStability`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

Use the instrument reported stability flag for logged data exclusion

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn`

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/StrictMode

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_ccn

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

DMT PAX

/<PROFILE>/Components/<IDENTIFIER>/ReportInterval

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/StrictMode

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Measurement wavelength

Type

Real number

Valid range

`0.0 < value`

Units

nm

Undefined values

532

Spancheck

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The time that air is flushed through the instrument before measurements begin

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

2 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`

- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The maximum amount of time measurements are taken during air gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Count`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Align`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The minimum amount of time measurements are taken during the air phase

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

10 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The number of units to apply in time

Type

[Integer](#)

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The time the span gas is flushed through the instrument before measurements begin

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

10 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Count

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Align

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax

Description

The maximum amount of time measurements are taken during the gas phase

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The number of units to apply in time

Type

[Integer](#)

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Align`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The minimum amount of time measurements are taken during the gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

5 minutes

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Units`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi

- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Type`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The type of span gas used

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- CO2 - Carbon dioxide, 2.61 times air Rayleigh scattering
- FM200 - Aflurane (C₂HF₅), 15.3 times air Rayleigh scattering

- SF6 - Sulfur hexafluoride, 6.74 times air Rayleigh scattering
- R12 - Freon-12 (CCl₂F₂), 15.31 times air Rayleigh scattering
- R22 - Chlorodifluoromethane (CHClF₂), 7.53 times air Rayleigh scattering
- R134 - Norflurane (CH₂FCF₃), 7.35 times air Rayleigh scattering

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Pressure`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The smoother applied to pressure measurements

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Scattering`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The smoother applied to angular scattering measurements

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Temperature`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The smoother applied to temperature measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/Commands`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/InstrumentCommands`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/TerminalAngle`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_dmt_pax`

Description

The maximum angle of integration for the spancheck

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 360.0$

Units

degrees

Undefined values

Not permitted

Ecotech Aurora 3000/4000 Nephelometer

`/<PROFILE>/Components/<IDENTIFIER>/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Instrument address

Type

Integer

Valid range

$0 \leq \text{value} \leq 6$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/BlankTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The time after a zero that data are ignored

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

62 seconds

`</PROFILE>/Components/<IDENTIFIER>/BlankTime/Units`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`</PROFILE>/Components/<IDENTIFIER>/BlankTime/Count`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/BlankTime/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/BurstDelay`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Delay between commands when interrogating

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/DataSmoothing`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Data smoothing used when the instrument requires smoothed data

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Kalman or KalmanFilter
- Average or MovingAverage
- None or Disable - Disable all instrument level smoothing

/<PROFILE>/Components/<IDENTIFIER>/InterrogationDelay

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

Delay between interrogation bursts

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/NormalizationTemperature

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

STP temperature set on the instrument, if available

Type

Real number

Units

°C

Undefined values

Ambient

/<PROFILE>/Components/<IDENTIFIER>/OpticalUpdateTimeout

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

Maximum time between optical (scattering) change detections before outputting anyway

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 120.0$

Units

s

Undefined values

Automatically calculated

/<PROFILE>/Components/<IDENTIFIER>/ReferenceUpdateTimeout**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

Maximum time reference intensity change detections before outputting anyway

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 120.0$

Units

s

Undefined values

Automatically calculated

/<PROFILE>/Components/<IDENTIFIER>/Retry/BusyTimeout**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

Maximum time to accept busy responses for before giving up

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Retry/Delay**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

Time to wait after a command failure before retrying it

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Retry/MaximumFailures`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Maximum number of failed or invalid (non-busy) command responses to accept before giving up

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/B`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The shortest (blue) wavelength of measurement

Type

Real number

Valid range

$0.0 < \text{value}$

Units

nm

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Wavelength/G**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

The middle (green) wavelength of measurement

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Wavelength/R**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

The longest (red) wavelength of measurement

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Zero/DisableCommands**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

Commands sent to the system to disable zero mode

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Zero/DisableInstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Commands sent to instruments to disable zero mode

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Zero/DisableInstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Zero/EnableCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Commands sent to the system to enable zero mode

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Zero/EnableInstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Commands sent to instruments to enable zero mode

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Zero/EnableInstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Mode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Zero calculation mode

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Offset` or `Software` - Software zero offset with no instrument on board adjustment
- `EnableFilter` - Use instrument native zeroing, but enable the filter only while in zero mode
- `Native` or `Instrument` - Instrument native zeroing

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Fill`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`
- `/<PROFILE>/Components/<IDENTIFIER>/Zero/Mode = Offset`

Description

The blank time before zero measurement to allow the neph to fill with filtered air

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

240 seconds

`<PROFILE>/Components/<IDENTIFIER>/Zero/Fill/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`
- `<PROFILE>/Components/<IDENTIFIER>/Zero/Mode = Offset`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`<PROFILE>/Components/<IDENTIFIER>/Zero/Fill/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`
- `<PROFILE>/Components/<IDENTIFIER>/Zero/Mode = Offset`

Description

The number of units to apply in time

Type

[Integer](#)

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Zero/Fill/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name` = `acquire_ecotech_nephaurora`
- `<PROFILE>/Components/<IDENTIFIER>/Zero/Mode` = `Offset`

Description

Align units in logical time

Type

[Boolean](#), values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Components/<IDENTIFIER>/Zero/Measure`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name` = `acquire_ecotech_nephaurora`
- `<PROFILE>/Components/<IDENTIFIER>/Zero/Mode` = `Offset`

Description

The time to measure filtered air

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

300 seconds

`<PROFILE>/Components/<IDENTIFIER>/Zero/Measure/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name` = `acquire_ecotech_nephaurora`
- `<PROFILE>/Components/<IDENTIFIER>/Zero/Mode` = `Offset`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`

- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Measure/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`
- `/<PROFILE>/Components/<IDENTIFIER>/Zero/Mode = Offset`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Measure/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`
- `/<PROFILE>/Components/<IDENTIFIER>/Zero/Mode = Offset`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Spancheck

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/InstrumentCommands`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/Commands`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/InstrumentCommands`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected

during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The time that air is flushed through the instrument before measurements begin

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

8 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`

- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Count`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Align`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The maximum amount of time measurements are taken during air gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Units

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Count

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Align

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

Align units in logical time

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample

Required conditions

- **</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora**

Description

The minimum amount of time measurements are taken during the air phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

10 minutes

</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Units

Required conditions

- **</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora**

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- **Millisecond or Milliseconds**
- **Second, Sec, Seconds, Secs, or s**
- **Minute, Min, Mins, Minutes, m, or mi**
- **Hour, Hours, or h**
- **Day, Days, or d**
- **Week, Weeks, or w**
- **Month, Mon, Mons, Months, or mo**
- **Quarter, Qtr, Qtrs, Quarters, or q**
- **Year, Years, or y**

</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Count

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The time the span gas is flushed through the instrument before measurements begin

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

8 minutes

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Units

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Count

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Align

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

The maximum amount of time measurements are taken during the gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Units**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The minimum amount of time measurements are taken during the gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

10 minutes

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Count`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Align`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Type`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The type of span gas used

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- **CO2** - Carbon dioxide, 2.61 times air Rayleigh scattering
- **FM200** - Aflurane (C₂HF₅), 15.3 times air Rayleigh scattering
- **SF6** - Sulfur hexafluoride, 6.74 times air Rayleigh scattering
- **R12** - Freon-12 (CCl₂F₂), 15.31 times air Rayleigh scattering
- **R22** - Chlorodifluoromethane (CHClF₂), 7.53 times air Rayleigh scattering
- **R134** - Norflurane (CH₂FCF₃), 7.35 times air Rayleigh scattering

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Dark`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The smoother applied to dark count rates

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Measurement`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The smoother applied to measurement count rates

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the

Type [enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Pressure`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The smoother applied to pressure measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Reference`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The smoother applied to reference count rates

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Scattering`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The smoother applied to angular scattering measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Temperature`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The smoother applied to temperature measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/TerminalAngle`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_ecotech_nephaurora`

Description

The maximum angle of integration for the spancheck

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 360.0$

Units

degrees

Undefined values

Not permitted

Eigenbrodt NMO191 Rain Sampler

`/<PROFILE>/Components/<IDENTIFIER>/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_eigenbrodt_nmo191`

Description

Instrument address number

Type

Integer

Undefined values

Any

/<PROFILE>/Components/<IDENTIFIER>/PollInterval**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_eigenbrodt_nmo191

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/StrictMode**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_eigenbrodt_nmo191

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Generic METAR conversion

/<PROFILE>/Components/<IDENTIFIER>/Altitude**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_metar

Description

The station altitude used in pressure conversion calculations

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

m

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/BaseTime`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_metar`

Description

Absolute time origin used when interpreting logged data

Type

Real number

Units

s

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/BaseTime`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_metar`

Description

Absolute time origin used when interpreting logged data

Structure

This is an absolute point in time specified as the number of seconds from the Unix epoch (1970-01-01T00:00:00Z).

`</PROFILE>/Components/<IDENTIFIER>/ReportEveryRecord`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_metar`

Description

Output data for every METAR record instead of only ones with reported time elapsed

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

<PROFILE>/Components/<IDENTIFIER>/ReportInterval

Required conditions

- **<PROFILE>/Components/<IDENTIFIER>/Name** = acquire_generic_metar

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

Generic passive data conversion

<PROFILE>/Components/<IDENTIFIER>/AllowMultipleMatches

Required conditions

- **<PROFILE>/Components/<IDENTIFIER>/Name** = acquire_generic_passive

Description

Allow the same input line to match multiple record definitions

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

<PROFILE>/Components/<IDENTIFIER>/AllowUnmatchedLines

Required conditions

- **<PROFILE>/Components/<IDENTIFIER>/Name** = acquire_generic_passive

Description

Allow an input line to not match any record definition without it being considered invalid

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

<PROFILE>/Components/<IDENTIFIER>/InstrumentMetadata

Required conditions

- **<PROFILE>/Components/<IDENTIFIER>/Name** = acquire_generic_passive

Description

The base metadata that the instrument is associated with

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/InstrumentMetadata/Manufacturer`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The instrument manufacturer

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/InstrumentMetadata/Model`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The instrument model identifier

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/InstrumentMetadata/SerialNumber`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The instrument serial number

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Records`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The records the generic processor accepts

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Fields/Quote`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The field quotation indicator

Type

Text

Undefined values

"

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Fields/QuoteEscape`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The field quotation escape from within a quoted field

Type

Text

Undefined values

""

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Fields/Separator`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The separator between fields

Type

Text

Undefined values

,

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Header/Match`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The regular expression that a header must match

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Header/MatchCaseInsensitive`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Perform case-insensitive matching

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Interval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The expected interval between records

Type

Real number

Valid range

0.0 < value

Units

s

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Match`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The regular expression any line must match to be accepted

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/MatchCaseInsensitive`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Perform case-insensitive matching

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/RequireValidVariable`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Require at least one non-missing variable for the record to be accepted

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Rounding`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The maximum time to shift records to account for the expected time interval rounding

Type

Real number

Valid range

0.0 < value

Units

s

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Substitutions`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Line replacement operations

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Substitutions/#0/CaseInsensitive`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Match the string in case insensitive mode

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Substitutions/#0/Match`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Input string to match

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Substitutions/#0/Replacement`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Replacement string with pattern captures available

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Time/Fields`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The fields used to parse the time

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Time/Fields/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The indices of the fields used to generate a time parsing list

Type

Integer

Valid range

1 ≤ value

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Time/RequireValid**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive

Description

Require a valid time the record to be accepted

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive

Description

Defined output variables

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed <IDENTIFIER> is used as a placeholder to real name selected during configuration.

/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Accumulation**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive

Description

The accumulation to rate conversion behavior

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Disabled or None - No accumulation to rate conversion is performed for the variable
- Rate - Convert the value to a rate by taking the difference between the current input and the prior, then dividing by the time between the two readings
- Absolute or Difference - Convert the value by taking the difference between the current

reading and the prior one

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Calibration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Calibration applied to the raw input value

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with [real number](#) values. For example, `#0` is the constant offset while `#1` is the slope.

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Captures`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The matching captures to look for a value, in order

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Captures/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The indices of the captures from the matching regular expression used to select the value from

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Fields`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The fields to look for a value, in order

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Fields/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The indices of the fields used to select the value from

Type

Integer

Valid range

$1 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/HeaderCaseInsensitive`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Perform case-insensitive matching for header matching

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Headers`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The matching header patterns to look for a value, in order

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths

represented here use #0 as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Headers/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The regular expressions to match against headers used to select the value from

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/MVC`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

A regular expression used to check if the input is considered missing

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/MVCBehavior`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The behavior applied when the variable is present but matches the explicitly missing regular expression

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `MVC`, `OutputInvalid`, or `OutputMVC` - Output a specifically invalidated value for the variable
- `Invalid` or `InvalidRecord` - Consider the entire record invalid
- `Gap`, `IgnoreWithGap`, or `SkipWithGap` - Introduce an explicit gap in the output for the variable
- `Ignore` or `Skip` - Ignore the invalid value and continue reporting the prior output

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/MVCCaseInsensitive`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Perform case-insensitive matching for missing input checking

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/MaximumAge`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The maximum time between variable updates before it is considered a gap

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Metadata`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Base metadata for the variable

Structure

This value represents metadata injected directly into some output. That is, the exact contents defined by this value are used as metadata for another value. The exact keys are determined by what metadata is required.

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Metadata/*dDescription`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

A sort description of the variable meaning

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Metadata/*dGroupUnits`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The units the variable is groups with for display

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Metadata/*dNoteCalibration`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Any notes about the calibration (location, date, etc)

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Metadata/*dUnits`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The physical units the variable is reported in

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/MissingBehavior`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The behavior applied when the variable is missing from the record

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Invalid` or `InvalidRecord` - Consider the entire record invalid
- `MVC`, `OutputInvalid`, or `OutputMVC` - Output a specifically invalidated value for the variable

- `Gap`, `IgnoreWithGap`, or `SkipWithGap` - Introduce an explicit gap in the output for the variable
- `Ignore` or `Skip` - Ignore the invalid value and continue reporting the prior output

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/Name`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Output variable name

Type

Text

Undefined values

Key value

`/<PROFILE>/Components/<IDENTIFIER>/Records/#0/Variables/<IDENTIFIER>/SearchBehavior`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

The value searching behaviour for the variable input

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `FirstPresent` or `FirstNonEmpty` - Use the first value available that is not an empty field
- `Defined` or `FirstValid` - Use the first value that does not match the missing value matcher

`/<PROFILE>/Components/<IDENTIFIER>/Timeout`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_generic_passive`

Description

Timeout before communications loss is reported

Type

Real number

Valid range

0.0 < value

Units

Undefined values

Not permitted

GMD CLAP-3W or BMI TAP

`/<PROFILE>/Components/<IDENTIFIER>/Area`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The areas of all eight spots on the clap filter

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices. The array is expected to contain 8 values.

`/<PROFILE>/Components/<IDENTIFIER>/Area/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The area in mm² of a single sampling spot

Type

Real number

Valid range

0.1 ≤ value

Units

mm²

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Autodetect/End/Band`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The relative scaling band for intensities relative to a white filter that must be met before a filter change end is detected

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Autodetect/End/Enable`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

Enable end of filter change auto-detection

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Autodetect/Start/Dark`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The threshold that triggers a filter change start when any of the dark intensities exceeds it

Type

Real number

Valid range

$0.0 < \text{value}$

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Autodetect/Start/Enable`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

Enable start of filter change auto-detection

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Autodetect/Start/Intensity`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The threshold that triggers a filter change start when any of the sample intensities is below it

Type

Real number

Valid range

$0.0 < \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Autodetect/Start/RequiredTriggers`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The number of simultaneous intensity triggers required to trigger start autodetection

Type

Integer

Valid range

$1 \leq \text{value}$

Undefined values

2

`/<PROFILE>/Components/<IDENTIFIER>/BypassResumeContaminate`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The time to flag data as contaminated after resuming from a bypass

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

30 seconds aligned

`/<PROFILE>/Components/<IDENTIFIER>/BypassResumeContaminate/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`/<PROFILE>/Components/<IDENTIFIER>/BypassResumeContaminate/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/BypassResumeContaminate/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

Align units in logical time

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Components/<IDENTIFIER>/CheckInstrumentTiming`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

Enable verification of the instrument clock with the computer clock by checking the number of records in an hour

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Components/<IDENTIFIER>/Filter/Baseline`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The smoother applied to the filter readings at the start of a filter to determine stability and the initial parameters

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the **Type enumeration** key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

A 90 second boxcar with a maximum RSD of 0.001

`<PROFILE>/Components/<IDENTIFIER>/Filter/CheckInstrumentID`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

Check the reported instrument identifier against the saved state before resuming a filter after a restart

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Components/<IDENTIFIER>/Filter/VerifyWhiteBand`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The relative scaling band to confirm a white filter after a change

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/FlowScale`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The scaling factor applied to the calibrated flow and volumes to get the final calibrated values

Type

Real number

Valid range

$0.0 < \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/HardwareFlowCalibration`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The internal voltage to lpm hardware flow calibration

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with [real number](#) values. For example, `#0` is the constant offset while `#1` is the slope.

`/<PROFILE>/Components/<IDENTIFIER>/IntegrateVolume`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

Enable regeneration of the total volume by integrating the flow in software

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/RealtimeDiagnostics**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w

Description

Enable all realtime data diagnostic generation, resulting in a greatly increased amount of data generated

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Spot/AdvanceTransmittance/B**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w

Description

The blue transmittance below which an automatic spot advance is initiated

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

0.5

/<PROFILE>/Components/<IDENTIFIER>/Spot/AdvanceTransmittance/G**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w

Description

The green transmittance below which an automatic spot advance is initiated

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

0.7

/<PROFILE>/Components/<IDENTIFIER>/Spot/AdvanceTransmittance/R

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The red transmittance below which an automatic spot advance is initiated

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

0.5

`/<PROFILE>/Components/<IDENTIFIER>/Spot/Normalize`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The smoother applied to the spot readings when initializing to determine stability and set the normalization for transmittance 1.0

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

A 60 second boxcar, discarding the first 8 seconds, and requiring at least thirty, with a maximum RSD of 0.001

`/<PROFILE>/Components/<IDENTIFIER>/Spot/ResumeTimeout`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The maximum time after an interruption of data collection to accept the active spot instead of advancing to the next one

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

30 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spot/ResumeTimeout/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`/<PROFILE>/Components/<IDENTIFIER>/Spot/ResumeTimeout/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spot/ResumeTimeout/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

Align units in logical time

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/B`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The shortest (blue) wavelength of measurement

Type

Real number

Valid range

$0.0 < \text{value}$

Units

nm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/G`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The middle (green) wavelength of measurement

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/R`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_clap3w`

Description

The longest (red) wavelength of measurement

Type

[Real number](#)

Valid range

0.0 < value

Units

nm

Undefined values

Not permitted

GMD CPC pulse counter box

`/<PROFILE>/Components/<IDENTIFIER>/Channel`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_cpcpulse`

Description

Sampling channel

Type

[Integer](#)

Valid range

$1 \leq \text{value} \leq 2$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Flow`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_cpcpulse`

Description

Counter sample flow rate

Type

Real number

Valid range

0.0 < value

Units

lpm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/UseMeasuredTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_gmd_cpcpulse`

Description

Use the measured time between records for count rate calculations instead of assuming 1 Hz

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Grimm OPC Models 110x

`/<PROFILE>/Components/<IDENTIFIER>/Firmware`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_grimm_opc110x`

Description

The assumed firmware code used for mass calculations if one cannot be determined from the instrument

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Model`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_grimm_opc110x`

Description

The assumed model number used for bin numbering if one cannot be determined from the

instrument

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_grimm_opc110x`

Description

This causes the system to refuse records that do not match known types

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/UseMeasuredTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_grimm_opc110x`

Description

Use the measured time between records for count rate calculations instead of assuming 1/6 Hz

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Love PID Controller

`/<PROFILE>/Components/<IDENTIFIER>/Controllers`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Fixed controller definitions

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Controllers/#0/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Controller address

Type

Integer

Valid range

$1 \leq \text{value} \leq 1023$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Controllers/#0/Bypass**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid

Description

Setpoint output applied on system bypass

Type

Real number

/<PROFILE>/Components/<IDENTIFIER>/Controllers/#0/DecimalDigits**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid

Description

Number of places after the decimal point to set the controller to

Type

Integer

Valid range

$0 \leq \text{value} \leq 3$

Undefined values

Unchanged

/<PROFILE>/Components/<IDENTIFIER>/Controllers/#0/ManualMode**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid

Description

Enable explicit manual control mode

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Controllers/#0/Setpoint**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Setpoint output applied on initialization

Type

Real number

`/<PROFILE>/Components/<IDENTIFIER>/Controllers/#0/Shutdown`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Setpoint output applied on system shutdown

Type

Real number

`/<PROFILE>/Components/<IDENTIFIER>/Controllers/#0/UnBypass`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Setpoint output applied on system un-bypass

Type

Real number

`/<PROFILE>/Components/<IDENTIFIER>/PollInterval`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

1.0

`/<PROFILE>/Components/<IDENTIFIER>/ProvisionalTimeout`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Maximum time to wait for implicitly created controllers before giving up

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

120.0

`/<PROFILE>/Components/<IDENTIFIER>/RetryTimes`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Maximum number of times to retry commands before failure

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/SetpointOutputs`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Output name to controller setpoint address mappings

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/SetpointOutputs/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Controller address

Type

Integer

Valid range

$1 \leq \text{value} \leq 1023$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Variables`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Defined output variables

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Controller address

Type

Integer

Valid range

$1 \leq \text{value} \leq 1023$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Calibration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Calibration applied to the raw input value

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with [real number](#) values. For example, `#0` is the constant offset while `#1` is the slope.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Base metadata for the variable

Structure

This value represents metadata injected directly into some output. That is, the exact contents defined by this value are used as metadata for another value. The exact keys are determined by what metadata is required.

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dDescription`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

A sort description of the variable meaning

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dGroupUnits`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

The units the variable is groups with for display

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dNoteCalibration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Any notes about the calibration (location, date, etc)

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Metadata/*dUnits`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

The physical units the variable is reported in

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Variables/<IDENTIFIER>/Name`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_love_pid`

Description

Output variable name

Type

Text

Undefined values

Key value

Magee Aethalometer Model AE33

`/<PROFILE>/Components/<IDENTIFIER>/ATNf1`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The lower attenuation threshold to use and set on the instrument used to determine the start of FVRF calculation

Type

Real number

Valid range

0.0 < value

Undefined values

Read on startup

`/<PROFILE>/Components/<IDENTIFIER>/ATNf2`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The second attenuation threshold to use and set on the instrument used to determine the end of FVRF calculation and the beginning of new-constant application

Type

Real number

Valid range

0.0 < value

Undefined values

Read on startup

`/<PROFILE>/Components/<IDENTIFIER>/AbsorptionEfficiency`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

Absorption efficiency of the measurement channels

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices. The array is expected to contain 7 values.

`/<PROFILE>/Components/<IDENTIFIER>/AbsorptionEfficiency/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The absorption efficiency of the channel

Type

Real number

Valid range

0.0 < value

Units

m²/g

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Area`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The areas of both sampling spots

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices. The array is expected to contain 2 values.

`/<PROFILE>/Components/<IDENTIFIER>/Area/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The area in mm² of a single sampling spot

Type

Real number

Valid range

0.1 ≤ value

Units

mm²

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/DisableEBCZeroCheck`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

This disables the check for all EBCs being exactly zero to indicate a tape advance in progress

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

/<PROFILE>/Components/<IDENTIFIER>/Leakage

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33

Description

The leakage factor (ζ) to use and set on the instrument

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

Read on startup

/<PROFILE>/Components/<IDENTIFIER>/RealtimeDiagnostics

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33

Description

Enable all realtime data diagnostic generation, resulting in a greatly increased amount of data generated

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/SetInstrumentTime

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33

Description

Set the internal clock on the instrument on start up, this can cause faults on the instrument requiring power cycles

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Spot1FlowCalibration

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33

Description

The calibration applied to the flow through spot one

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with **real number** values. For example, **#0** is the constant offset while **#1** is the slope.

/<PROFILE>/Components/<IDENTIFIER>/Spot2FlowCalibration

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33

Description

The calibration applied to the flow through spot two

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with **real number** values. For example, **#0** is the constant offset while **#1** is the slope.

/<PROFILE>/Components/<IDENTIFIER>/StrictMode

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/TimeBase

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33

Description

The instrument calculation and averaging time

Type

Integer

Valid range

$1 \leq \text{value}$

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/TotalFlowCalibration`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The calibration applied to the combined total flow

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with [real number](#) values. For example, `#0` is the constant offset while `#1` is the slope.

`<PROFILE>/Components/<IDENTIFIER>/UseMeasuredTime`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The behavior used to determine the elapsed time between reports for integration

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Rounded` - Use the measured time between reports rounded to the nearest multiple of the reported time base
- `Reported` - Use the instrument reported time base
- `Measured` - Use the exact measured time between reports

`<PROFILE>/Components/<IDENTIFIER>/Wavelengths`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The wavelengths of the measurement channels

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices. The array is expected to contain 7

values.

`/<PROFILE>/Components/<IDENTIFIER>/Wavelengths/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The central measurement wavelength of the channel

Type

Real number

Valid range

$0.0 < \text{value}$

Units

nm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/WeingartnerConstant`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The Weingartner constant © to use and set on the instrument

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

Read on startup

`/<PROFILE>/Components/<IDENTIFIER>/kMax`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The upper bound of accepted correction constants to use and set on the instrument

Type

Real number

Undefined values

Read on startup

`/<PROFILE>/Components/<IDENTIFIER>/kMin`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer33`

Description

The lower bound of accepted correction constants to use and set on the instrument

Type

Real number

Undefined values

Read on startup

Magee Aethalometer Models AE16/AE21/AE31

`/<PROFILE>/Components/<IDENTIFIER>/AbsorptionEfficiency`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131`

Description

Absorption efficiency of the measurement channels

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices. The array is expected to contain 7 values.

`/<PROFILE>/Components/<IDENTIFIER>/AbsorptionEfficiency/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131`

Description

The absorption efficiency of the channel

Type

Real number

Valid range

0.0 < value

Units

m²/g

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Area`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131`

Description

The area in mm² of the sampling spot

Type

Real number

Valid range

0.1 ≤ value

Units

mm²

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/BCUnits`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131`

Description

The expected units of the reported EBCs

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Automatic` or `Auto` - Attempt to determine the units automatically by comparing the EBCs to the acquisition system's calculations
- `ug`, `u`, `ug/m3`, or `ugm3` - Assume the reported EBCs are in µg/m³
- `ng`, `n`, `ng/m3`, or `ngm3` - Assume the reported EBCs are in ng/m³

`<PROFILE>/Components/<IDENTIFIER>/FilterChangeDetectThreshold`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131`

Description

The minimum transmittance change required to detect a spot advance

Type

Real number

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/FlowCalibration**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131

Description

The calibration applied to the sample flow rate

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with **real number** values. For example, #0 is the constant offset while #1 is the slope.

/<PROFILE>/Components/<IDENTIFIER>/IgnoreFraction**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131

Description

Ignore the bypass fraction reported by the instrument and assume 1.0 (no air bypassed)

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/MeanRatio**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131

Description

The mean ratio multiplication factor

Type

Real number

Valid range

0.0 < value

Undefined values

Not permitted

<PROFILE>/Components/<IDENTIFIER>/ReportInterval

Required conditions

- <PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

<PROFILE>/Components/<IDENTIFIER>/SamplePressure

Required conditions

- <PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131

Description

The sample pressure setting on the instrument

Type

Real number

Units

hPa

Undefined values

Not permitted

<PROFILE>/Components/<IDENTIFIER>/SampleTemperature

Required conditions

- <PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131

Description

The sample temperature setting on the instrument

Type

Real number

Units

°C

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/StrictMode**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

/<PROFILE>/Components/<IDENTIFIER>/Wavelengths**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131

Description

The wavelengths of the measurement channels

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, **#0** is the first element of the array. The paths represented here use **#0** as a placeholder for all indices. The array is expected to contain 7 values.

/<PROFILE>/Components/<IDENTIFIER>/Wavelengths/#0**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_aethalometer162131

Description

The central measurement wavelength of the channel

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

Not permitted

Magee TCA-08

`<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_tca08`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_tca08`

Description

The expected report update time

Type

Real number

Valid range

$0.0 < \text{value}$

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_magee_tca08`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

PurpleAir PA-II

`/<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_purpleair_pa2`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_purpleair_pa2`

Description

The maximum expected report update time

Type

Real number

Valid range

$0.0 < \text{value}$

Units

s

Undefined values

Not permitted

R.M. Young 86xxx Sonic Anemometer

`/<PROFILE>/Components/<IDENTIFIER>/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rmy_wind86xxx`

Description

Instrument address

Type

Text

Undefined values

Any

`/<PROFILE>/Components/<IDENTIFIER>/ReportInterval`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rmy_wind86xxx`

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/StrictMode`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rmy_wind86xxx`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Radiance Research Nephelometer Model M903

`/<PROFILE>/Components/<IDENTIFIER>/Relay/Spancheck`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Relay engaged during gas spancheck measurements

Type

Integer

Valid range

$1 \leq \text{value} \leq 2$

/<PROFILE>/Components/<IDENTIFIER>/Relay/Zero**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

Relay engaged during zero measurements

Type

Integer

Valid range

$1 \leq \text{value} \leq 2$

/<PROFILE>/Components/<IDENTIFIER>/ReportInterval**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

The expected un-pollled response time

Type

Real number

Valid range

$0.0 < \text{value}$

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/StrictMode**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Wavelength**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

Measurement wavelength

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

530

/<PROFILE>/Components/<IDENTIFIER>/Zero/Flush**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

The amount of time to flush air through the nephelometer before or after zeroing

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

62 seconds

/<PROFILE>/Components/<IDENTIFIER>/Zero/Flush/Units**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

/<PROFILE>/Components/<IDENTIFIER>/Zero/Flush/Count

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Zero/Flush/Align

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Zero/Sample

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

The amount of time to sample and average air when zeroing

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

5 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Sample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Sample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Sample/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Spancheck

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The time that air is flushed through the instrument before measurements begin

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

2 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The maximum amount of time measurements are taken during air gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The minimum amount of time measurements are taken during the air phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

10 minutes

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Count`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Align`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/End/Commands`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The time the span gas is flushed through the instrument before measurements begin

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

10 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The maximum amount of time measurements are taken during the gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Count`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Align`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The minimum amount of time measurements are taken during the gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

5 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Type`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The type of span gas used

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `C02` - Carbon dioxide, 2.61 times air Rayleigh scattering
- `FM200` - Apaflurane (C₂HF₂), 15.3 times air Rayleigh scattering
- `SF6` - Sulfur hexafluoride, 6.74 times air Rayleigh scattering
- `R12` - Freon-12 (CCl₂F₂), 15.31 times air Rayleigh scattering
- `R22` - Chlorodifluoromethane (CHClF₂), 7.53 times air Rayleigh scattering
- `R134` - Norflurane (CH₂FCF₃), 7.35 times air Rayleigh scattering

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Dark`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The smoother applied to dark count rates

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Measurement

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

The smoother applied to measurement count rates

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Pressure

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

The smoother applied to pressure measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Reference

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903

Description

The smoother applied to reference count rates

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Scattering`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The smoother applied to angular scattering measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Temperature`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The smoother applied to temperature measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/TerminalAngle`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_neph903`

Description

The maximum angle of integration for the spancheck

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 360.0$

Units

degrees

Undefined values

Not permitted

Radiance Research 1-W PSAP

`/<PROFILE>/Components/<IDENTIFIER>/Area`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

The area in mm² of the sampling spot

Type

Real number

Valid range

$0.1 \leq \text{value}$

Units

mm²

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Autodetect/End/Enable`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

Enable end of filter change auto-detection

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Autodetect/Smoother`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

The smoother applied to the filter readings for determining auto-detection events

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable. The smoother is used for spike detection. This means that the type of smoother selected should provide an indicator of when the incoming value is considered a spike.

Undefined values

A 60 second boxcar, requiring at least thirty second of data, with a maximum RSD of 0.02 and a spike band of 2.0

`<PROFILE>/Components/<IDENTIFIER>/Autodetect/Start/Enable`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

Enable start of filter change auto-detection

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Filter/EnableNormalizationRecovery`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

Enable reconstruction of the normalization based on the reported transmittance

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Filter/ResumeTimeout`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

The maximum time after an interruption of data collection to accept the saved normalization parameters

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

2 hours

`/<PROFILE>/Components/<IDENTIFIER>/Filter/ResumeTimeout/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`/<PROFILE>/Components/<IDENTIFIER>/Filter/ResumeTimeout/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Filter/ResumeTimeout/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Filter/VerifyNormalizationBand**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w

Description

The relative scaling band to verify reference intensities of a saved normalization

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Filter/VerifyWhiteBand**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w

Description

The relative scaling band to confirm a white filter after a change

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/FlowCalibration**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w

Description

The calibration applied to the sample flow rate

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with **real number** values. For example, **#0** is the constant offset while **#1** is the slope.

`<PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`<PROFILE>/Components/<IDENTIFIER>/UseMeasuredTime`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

Use the measured time between records for integration instead of assuming 1 Hz

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`<PROFILE>/Components/<IDENTIFIER>/Wavelength`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap1w`

Description

Measurement wavelength

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

574

Radiance Research 3-W PSAP

`<PROFILE>/Components/<IDENTIFIER>/Absorption/EnableRecovery`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Use the reported absorptions when calculation is not possible

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Absorption/UseReported`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Use the reported absorption coefficients instead of recalculating them

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Area`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The area in mm² of the sampling spot

Type

Real number

Valid range

$0.1 \leq \text{value}$

Units

mm²

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Autodetect/End/Enable`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Enable end of filter change auto-detection

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`</PROFILE>/Components/<IDENTIFIER>/Autodetect/Smoother`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The smoother applied to the filter readings for determining auto-detection events

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the **Type enumeration** key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable. The smoother is used for spike detection. This means that the type of smoother selected should provide an indicator of when the incoming value is considered a spike.

Undefined values

A 60 second boxcar, requiring at least thirty second of data, with a maximum RSD of 0.02 and a spike band of 2.0

`</PROFILE>/Components/<IDENTIFIER>/Autodetect/Start/Enable`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Enable start of filter change auto-detection

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`<PROFILE>/Components/<IDENTIFIER>/Filter/EnableNormalizationRecovery`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Enable reconstruction of the normalization based on the reported transmittance

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`<PROFILE>/Components/<IDENTIFIER>/Filter/ResumeTimeout`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The maximum time after an interruption of data collection to accept the saved normalization parameters

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

2 hours

`<PROFILE>/Components/<IDENTIFIER>/Filter/ResumeTimeout/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- **Millisecond** or **Milliseconds**
- **Second**, **Sec**, **Seconds**, **Secs**, or **s**
- **Minute**, **Min**, **Mins**, **Minutes**, **m**, or **mi**

- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`<PROFILE>/Components/<IDENTIFIER>/Filter/ResumeTimeout/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Filter/ResumeTimeout/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Filter/VerifyNormalizationBand`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The relative scaling band to verify reference intensities of a saved normalization

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Filter/VerifyWhiteBand`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The relative scaling band to confirm a white filter after a change

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/FlowCalibration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The calibration applied to the sample flow rate

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with real number values. For example, #0 is the constant offset while #1 is the slope.

`/<PROFILE>/Components/<IDENTIFIER>/InstrumentArea`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The spot area setting on the instrument

Type

Real number

Valid range

$0.1 \leq \text{value}$

Units

mm²

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/UseMeasuredTime`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Use the measured time between records for integration instead of assuming 1 Hz

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/UseReportedArea`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Use the reported spot area from the instrument during recovery

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/B`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The shortest (blue) wavelength of measurement

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/G`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The middle (green) wavelength of measurement

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/R`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The longest (red) wavelength of measurement

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Weiss/A`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The Weiss "A" constant set on the instrument

Type

Real number

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Weiss/B`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

The Weiss "B" constant set on the instrument

Type

Real number

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Weiss/Reverse`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Back out the Weiss correction when using reported absorptions

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Weiss/UseReported`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_rr_psap3w`

Description

Use the reported Weiss constants when available

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Thermo MAAP

`</PROFILE>/Components/<IDENTIFIER>/Address`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

Instrument address number

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Any

`</PROFILE>/Components/<IDENTIFIER>/Area`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The area in mm² of the sampling spot

Type

Real number

Valid range

$0.1 \leq \text{value}$

Units

mm²

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Autodetect/Enable`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

Enable spot or filter change auto-detection based on intensities instead of simple flagging

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/BCUnits`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The expected units of the reported EBCs

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Automatic` or `Auto` - Attempt to determine the units automatically by comparing the EBCs to the acquisition system's calculations
- `ug`, `u`, `ug/m3`, or `ugm3` - Assume the reported EBCs are in $\mu\text{g}/\text{m}^3$
- `ng`, `n`, `ng/m3`, or `ngm3` - Assume the reported EBCs are in ng/m^3

`<PROFILE>/Components/<IDENTIFIER>/Filter/CheckInstrumentID`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

Check the reported instrument identifier against the saved state before resuming a filter after a restart

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Filter/Smoother`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The smoother used for filter and spot change auto-detection

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable. The smoother is used for spike detection. This means that the type of smoother selected should provide an indicator of when the incoming value is considered a spike.

Undefined values

A 600 second boxcar, discarding the first 120 and requiring 120 minimum, with a stable RSD of 0.05 and a spike band of 1.5

`</PROFILE>/Components/<IDENTIFIER>/FlowCalibration`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The calibration applied to the sample flow rate

Type

[Calibration](#)

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with [real number](#) values. For example, `#0` is the constant offset while `#1` is the slope.

`</PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The expected un-pollled response time

Type

[Real number](#)

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Spot/AdvanceAtHour`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The hour of day (1-24) to advance the spot at

Type

Integer

Valid range

$1 \leq \text{value} \leq 24$

Undefined values

Disabled

`</PROFILE>/Components/<IDENTIFIER>/Spot/AdvanceHours`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The maximum time to sample on a spot before advancing to the next one

Type

Integer

Valid range

$1 \leq \text{value} \leq 100$

Undefined values

100

`</PROFILE>/Components/<IDENTIFIER>/Spot/AdvanceTransmittance`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The minimum transmittance set on the instrument before it will attempt to advance the spot

Type

Real number

Valid range

$0.0 \leq \text{value}$

Undefined values

0.7

`<PROFILE>/Components/<IDENTIFIER>/Spot/ResumeTimeout`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The maximum time after an interruption of data collection to accept the active spot parameters

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

30 minutes

`<PROFILE>/Components/<IDENTIFIER>/Spot/ResumeTimeout/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`/<PROFILE>/Components/<IDENTIFIER>/Spot/ResumeTimeout/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spot/ResumeTimeout/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/TargetFlow`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

Target flow rate set on the instrument

Type

Real number

Valid range

0.0 < value

Units

lpm

Undefined values

Unchanged

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_maap5012`

Description

Measurement wavelength

Type

Real number

Valid range

0.0 < value

Units

nm

Undefined values

670

Thermo Series 49 Ozone Monitor

`/<PROFILE>/Components/<IDENTIFIER>/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Instrument address number

Type

Integer

Valid range

0 ≤ value ≤ 126

Undefined values

0

`/<PROFILE>/Components/<IDENTIFIER>/CommandDelay`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Delay time after receiving a response before sending the next command

Type

Real number

Valid range

0.0 ≤ value

Units

s

Undefined values

Automatic

`<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Delay time between instrument polling requests and controls the instrument averaging time

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Flush`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The time the level is flushed through the instrument before measurements begin

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

2 minutes

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Flush/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Flush/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Flush/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Levels`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Ozonator control levels at each stage of the spancheck

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Levels/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Ozonator control level

Type

Integer

Valid range

$1 \leq \text{value} \leq 5$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/MaximumSample`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The maximum amount of time measurements are taken during level measurement

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/MaximumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/MaximumSample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/MaximumSample/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/MinimumSample`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The minimum amount of time measurements are taken during level measurement

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

12 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/MinimumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/MinimumSample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/MinimumSample/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Concentration`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The smoother applied to the ozone concentration

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Pressure`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The smoother applied to pressure measurements

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Temperature`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The smoother applied to temperature measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`<PROFILE>/Components/<IDENTIFIER>/Zero/Flush`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The time that zero air flushed through the instrument before measurements begin

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

2 minutes

`<PROFILE>/Components/<IDENTIFIER>/Zero/Flush/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of

possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Flush/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Flush/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Zero/MaximumSample`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The maximum amount of time measurements are taken during zero measurement

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

/<PROFILE>/Components/<IDENTIFIER>/Zero/MaximumSample/Units

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

/<PROFILE>/Components/<IDENTIFIER>/Zero/MaximumSample/Count

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Zero/MaximumSample/Align

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Align units in logical time

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Zero/MinimumSample`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The minimum amount of time measurements are taken during zero measurement

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

12 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Zero/MinimumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`

- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Zero/MinimumSample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Zero/MinimumSample/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Smoothing/Concentration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The smoother applied to the ozone concentration

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the Type [enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Smoothing/Pressure`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The smoother applied to pressure measurements

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`/<PROFILE>/Components/<IDENTIFIER>/Zero/Smoothing/Temperature`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49`

Description

The smoother applied to temperature measurements

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

Thermo Series 49iQ Ozone Monitor

`/<PROFILE>/Components/<IDENTIFIER>/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49iq`

Description

Instrument address number for RS485 access

Type

Integer

Valid range

$0 \leq \text{value} \leq 255$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49iq`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/TCPFraming`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_thermo_ozone49iq`

Description

Use MODBUS TCP framing instead of RS485 RTU

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

TSI CPC Models 302x

`/<PROFILE>/Components/<IDENTIFIER>/Flow`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc302x`

Description

Counter sample flow rate

Type

Real number

Valid range

0.0 < value

Units

lpm

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/UseMeasuredTime

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc302x

Description

Use the measured time between records for count rate calculations instead of the reported rate

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

TSI CPC Models 377x

/<PROFILE>/Components/<IDENTIFIER>/Flow

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc377x

Description

Counter sample flow rate

Type

Real number

Valid range

0.0 < value

Units

lpm

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/PollInterval

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc377x

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/SetInstrumentTime`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc377x`

Description

Set the internal clock on the instrument on start up, this can cause faults on instrument

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

TSI CPC Model 3010

`<PROFILE>/Components/<IDENTIFIER>/Flow`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3010`

Description

Counter sample flow rate

Type

Real number

Valid range

$0.0 < \text{value}$

Units

lpm

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/UseMeasuredTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3010`

Description

Use the measured time between records for count rate calculations instead of the reported rate

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

TSI CPC Model 3781

`/<PROFILE>/Components/<IDENTIFIER>/Flow`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3781`

Description

Counter sample flow rate

Type

Real number

Valid range

0.0 < value

Units

lpm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3781`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/SampleTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3781`

Description

Instrument data collection sampling time

Type

Real number

Valid range

$0.1 \leq \text{value} \leq 3600.0$

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3781`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

TSI CPC Model 3783

`/<PROFILE>/Components/<IDENTIFIER>/Flow`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3783`

Description

Counter sample flow rate

Type

Real number

Valid range

$0.0 < \text{value}$

Units

lpm

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/PollInterval**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3783

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 < \text{value}$

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/SampleTime**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3783

Description

Instrument data collection sampling time

Type

Real number

Valid range

$0.1 \leq \text{value} \leq 3600.0$

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/StrictMode**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_cpc3783

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

TSI Mass Flow Meter Models 4xxx

/<PROFILE>/Components/<IDENTIFIER>/BinaryFlowScale

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_mfm4xxx

Description

Scaling factor applied when operating in binary mode

Type

Real number

Valid range

0.0 < value

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/SampleTime

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_mfm4xxx

Description

Instrument data collection sampling time

Type

Real number

Valid range

0.001 ≤ value ≤ 1.0

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/StrictMode

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_mfm4xxx

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

TSI Nephelometer Model 3563

`/<PROFILE>/Components/<IDENTIFIER>/DisableSpancheckValve`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Disable control of the analog output during spanchecks

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/MaximumLampCurrent`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The maximum lamp current before the total power is reduced

Type

Real number

Valid range

0.0 < value

Units

A

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/B`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Blower power fraction

Type

Integer

Valid range

$0 \leq \text{value} \leq 255$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/H**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

Enable sample heater

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCI/High/ADC**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The high calibration ADC reading

Type

Integer

Valid range

$1 \leq \text{value} \leq 65535$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCI/High/Value**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The high calibration temperature

Type

Real number

Valid range

$0.1 \leq \text{value} \leq 400.0$

Units

K

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCI/Low/ADC`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The low calibration ADC reading

Type

Integer

Valid range

$0 \leq \text{value} \leq 65534$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCI/Low/Value`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The low calibration temperature

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 399.9$

Units

K

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCP/High/ADC`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The high calibration ADC reading

Type

Integer

Valid range

$1 \leq \text{value} \leq 65535$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCP/High/Value**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The high calibration pressure

Type

Real number

Valid range

$0.1 \leq \text{value} \leq 1200.0$

Units

hPa

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCP/Low/ADC**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The low calibration ADC reading

Type

Integer

Valid range

$0 \leq \text{value} \leq 65534$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCP/Low/Value**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The low calibration pressure

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 1199.9$

Units

hPa

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCR/High/ADC`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The high calibration ADC reading

Type

Integer

Valid range

$1 \leq \text{value} \leq 1023$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCR/High/Value`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The high calibration pressure

Type

Real number

Valid range

$0.1 \leq \text{value} \leq 100.0$

Units

%

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCR/Low/ADC

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The low calibration ADC reading

Type

Integer

Valid range

$0 \leq \text{value} \leq 1022$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCR/Low/Value

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The low calibration pressure

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 99.9$

Units

%

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCS/High/ADC

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The high calibration ADC reading

Type

Integer

Valid range

$1 \leq \text{value} \leq 65535$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCS/High/Value`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The high calibration temperature

Type

Real number

Valid range

$0.1 \leq \text{value} \leq 400.0$

Units

K

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCS/Low/ADC`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The low calibration ADC reading

Type

Integer

Valid range

$0 \leq \text{value} \leq 65534$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SCS/Low/Value`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The low calibration temperature

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 399.9$

Units

K

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SKB/K1`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Dead time of the PMT pulses

Type

Integer

Valid range

$1 \leq \text{value} \leq 65535$

Units

ps

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SKB/K2`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Total scatter calibration of the reference chopper

Type

Real number

Valid range

$0.0 < \text{value}$

Units

m⁻¹

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SKB/K3

Required conditions

- **/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563**

Description

Rayleigh scattering of air

Type

Real number

Valid range

$0.0 < \text{value}$

Units

m^{-1}

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SKB/K4

Required conditions

- **/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563**

Description

Backscatter calibration fraction

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 1.0$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SKG/K1

Required conditions

- **/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563**

Description

Dead time of the PMT pulses

Type

Integer

Valid range

$1 \leq \text{value} \leq 65535$

Units

ps

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Parameters/SKG/K2`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Total scatter calibration of the reference chopper

Type

Real number

Valid range

0.0 < value

Units

m⁻¹

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Parameters/SKG/K3`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Rayleigh scattering of air

Type

Real number

Valid range

0.0 < value

Units

m⁻¹

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Parameters/SKG/K4`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Backscatter calibration fraction

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 1.0$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SKR/K1`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Dead time of the PMT pulses

Type

Integer

Valid range

$1 \leq \text{value} \leq 65535$

Units

ps

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SKR/K2`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Total scatter calibration of the reference chopper

Type

Real number

Valid range

$0.0 < \text{value}$

Units

m⁻¹

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SKR/K3`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Rayleigh scattering of air

Type

Real number

Valid range

$0.0 < \text{value}$

Units

m^{-1}

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SKR/K4`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Backscatter calibration fraction

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 1.0$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Parameters/SL`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Calibration label

Type

Text

Undefined values

Unchanged

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SMB

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

Enable backscatter shutter

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SMZ

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

Auto-zero mode (0=manual, 1=single, 2-24=averaged)

Type

Integer

Valid range

$0 \leq \text{value} \leq 24$

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/SP

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

Lamp power

Type

Integer

Valid range

$0 \leq \text{value} \leq 150$

Units

w

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/STA

Required conditions

- **/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563**

Description

Averaging time

Type

Integer

Valid range

$1 \leq \text{value} \leq 9960$

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/STB

Required conditions

- **/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563**

Description

Blanking time

Type

Integer

Valid range

$15 \leq \text{value} \leq 999$

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Parameters/STP

Required conditions

- **/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563**

Description

Time between auto-zeros

Type

Integer

Valid range

$10 \leq \text{value} \leq 32767$

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Parameters/STZ`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Zero measurement duration

Type

Integer

Valid range

$1 \leq \text{value} \leq 9999$

Units

s

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Parameters/SVB`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Blue PMT voltage

Type

Integer

Valid range

$800 \leq \text{value} \leq 1200$

Units

V

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Parameters/SVG`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Green PMT voltage

Type

Integer

Valid range

$800 \leq \text{value} \leq 1200$

Units

V

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Parameters/SVR`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Red PMT voltage

Type

Integer

Valid range

$800 \leq \text{value} \leq 1200$

Units

V

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Parameters/V`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Zero valve control

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The expected un-pollled response time

Type

Real number

Valid range

$0.0 < \text{value}$

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/B`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The shortest (blue) wavelength of measurement

Type

Real number

Valid range

$0.0 < \text{value}$

Units

nm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/G`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The middle (green) wavelength of measurement

Type

Real number

Valid range

$0.0 < \text{value}$

Units

nm

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Wavelength/R`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The longest (red) wavelength of measurement

Type

Real number

Valid range

$0.0 < \text{value}$

Units

nm

Undefined values

Not permitted

Spancheck

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Activate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Deactivate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The time that air is flushed through the instrument before measurements begin

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

2 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Count

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/Flush/Align

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The maximum amount of time measurements are taken during air gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MaximumSample/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The minimum amount of time measurements are taken during the air phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

10 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Air/MinimumSample/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/End/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Activate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Deactivate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The time the span gas is flushed through the instrument before measurements begin

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

10 minutes

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Units

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Count

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Flush/Align

Required conditions

- /<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The maximum amount of time measurements are taken during the gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

Unlimited

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MaximumSample/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The minimum amount of time measurements are taken during the gas phase

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

5 minutes

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Count`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/MinimumSample/Align`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Gas/Type`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The type of span gas used

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- **CO2** - Carbon dioxide, 2.61 times air Rayleigh scattering
- **FM200** - Afaflurane (C₂HF₅), 15.3 times air Rayleigh scattering
- **SF6** - Sulfur hexafluoride, 6.74 times air Rayleigh scattering
- **R12** - Freon-12 (CCl₂F₂), 15.31 times air Rayleigh scattering
- **R22** - Chlorodifluoromethane (CHClF₂), 7.53 times air Rayleigh scattering
- **R134** - Norflurane (CH₂FCF₃), 7.35 times air Rayleigh scattering

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Dark`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The smoother applied to dark count rates

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the **Type enumeration** key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Measurement`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The smoother applied to measurement count rates

Type

Baseline Smoother

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the

Type [enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Pressure`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The smoother applied to pressure measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Reference`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The smoother applied to reference count rates

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`</PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Revolutions`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The smoother applied to chopper revolution measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Scattering`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The smoother applied to angular scattering measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`<PROFILE>/Components/<IDENTIFIER>/Spancheck/Smoothing/Temperature`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The smoother applied to temperature measurements

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type

of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

Indefinite

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/Start/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spancheck/TerminalAngle`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_tsi_neph3563`

Description

The maximum angle of integration for the spancheck

Type

Real number

Valid range

$0.0 \leq \text{value} \leq 360.0$

Units

degrees

Undefined values

Not permitted

Vaisala PWD Models x2

`/<PROFILE>/Components/<IDENTIFIER>/ID`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisala_pwdx2`

Description

The instrument identification code

Type

Text

Undefined values

Automatically determined

`/<PROFILE>/Components/<IDENTIFIER>/PollInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisala_pwdx2`

Description

Delay time between instrument polling requests

Type

Real number

Valid range

$0.0 \leq \text{value}$

Units

s

Undefined values

15.0

`/<PROFILE>/Components/<IDENTIFIER>/PresentWeather`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisala_pwdx2`

Description

The source for present weather data

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Automatic` or `Auto` - Select the the weather source based on the polling interval
- `15Minute` or `15Min` - Use the 15-minute reading
- `1Hour` or `Hour` - Use the one-hour reading
- `Instantaneous` or `Instant` - Use the instantaneous reading

`/<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisala_pwdx2`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Visibility`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisala_pwdx2`

Description

The source for visibility data

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Automatic` or `Auto` - Select the the visibility source based on the polling interval
- `10Minute` or `10Min` - Use the 10-minute visibility
- `1Minute`, `1Min`, or `Minute` - Use the one-minute visibility

Vaisala WMT700 Sonic Anemometer

`/<PROFILE>/Components/<IDENTIFIER>/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisla_wmt700`

Description

Instrument address

Type

Text

Undefined values

Any

`/<PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisla_wmt700`

Description

The expected un-pollled response time

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisla_wmt700`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

Vaisala WXT Models 5xx

`/<PROFILE>/Components/<IDENTIFIER>/Address`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisala_wxt5xx`

Description

Instrument address

Type

Integer

Valid range

$0 \leq \text{value} \leq 61$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Heating`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisala_wxt5xx`

Description

Enable instrument heating

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/ReportInterval`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisala_wxt5xx`

Description

The expected un-pollled response time

Type

Real number

Valid range

$0.0 < \text{value}$

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/StrictMode`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = acquire_vaisala_wxt5xx`

Description

This causes the system to refuse records that contain extra unrecognized fields

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

Accumulation Controller

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The defined accumulators

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/AccumulateWhileBypassed`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Continue accumulation even while in bypass mode

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Activate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Activate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Activate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Deactivate/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Commands sent to all components

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Deactivate/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Deactivate/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Index`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The output accumulator index

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Metadata`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Base metadata for the variable

Structure

This value represents metadata injected directly into some output. That is, the exact contents defined by this value are used as metadata for another value. The exact keys are determined by what metadata is required.

`/<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Metadata/*dDescription`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

A sort description of the variable meaning

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Metadata/*dGroupUnits`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The units the variable is groups with for display

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Metadata/*dUnits`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The physical units the variable is reported in

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Name`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Output variable name

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/Accumulators/#0/Rate`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The accumulation rate input, in units/second

Type

Dynamic Real Value Input

Structure

This is a specification of a single real valued input used in calculation. The specification generally takes the form of either a value from the data or a constant.

`<PROFILE>/Components/<IDENTIFIER>/Advance`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The time between accumulator advances

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

`<PROFILE>/Components/<IDENTIFIER>/Advance/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`<PROFILE>/Components/<IDENTIFIER>/Advance/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The number of units to apply in time

Type

[Integer](#)

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Advance/Align`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/BlankIndex`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The index to accumulate on when blanking

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Unchanged

`/<PROFILE>/Components/<IDENTIFIER>/BlankTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The amount of time at the start of accumulation to sample on the blank

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

`/<PROFILE>/Components/<IDENTIFIER>/BlankTime/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`</PROFILE>/Components/<IDENTIFIER>/BlankTime/Count`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/BlankTime/Align`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/BypassIndex`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The index to accumulate on when the system is bypassed

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Unchanged

`<PROFILE>/Components/<IDENTIFIER>/BypassOnEnd`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Put the system into bypass once sampling has completed

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/BypassWhileChanging`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Put the system into bypass while a change is in progress

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/EndIndex`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The index to accumulate on after the entire cycle finishes

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Unchanged

`<PROFILE>/Components/<IDENTIFIER>/MinimumTime`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The minimum time on an accumulator below which the advance time is extended

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

`/<PROFILE>/Components/<IDENTIFIER>/MinimumTime/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`/<PROFILE>/Components/<IDENTIFIER>/MinimumTime/Count`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/MinimumTime/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Processing/Additional`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Additional inputs requested for processing

Type

Sequence Selection

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

`<PROFILE>/Components/<IDENTIFIER>/Processing/Components`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Processing chain stages

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices.

`<PROFILE>/Components/<IDENTIFIER>/Processing/Components/#0/Name`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Data generating component name

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Processing/Input/End`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The end time to invoke the generator with

Type

Real number

Units

s

Undefined values

Not permitted

`</PROFILE>/Components/<IDENTIFIER>/Processing/Input/End`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The end time to invoke the generator with

Structure

This is an absolute point in time specified as the number of seconds from the Unix epoch (1970-01-01T00:00:00Z).

`</PROFILE>/Components/<IDENTIFIER>/Processing/Input/File`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Input file name used with the generator

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Processing/Input/IgnoreTimes`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Do not provide times to the input generator

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Merge`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Allow merging of identical processing inputs

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Name`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Data generating component name

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Start`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The start time to invoke the generator with

Type

Real number

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Start`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The start time to invoke the generator with

Structure

This is an absolute point in time specified as the number of seconds from the Unix epoch (1970-01-01T00:00:00Z).

`<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Station`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Input station used with the generator

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/Processing/Merge`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

Allow merging of identical processing chains

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Processing/MergeKey`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The unique identifier that must match for chain merging

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/StartIndex`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The index to begin sampling on

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/UpdateTimeout`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_accumulate`

Description

The maximum time to wait between data updates before values are invalidated

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

Generic Command Output

`/<PROFILE>/Components/<IDENTIFIER>/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Command definitions

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/After/BitClear`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Interpret state variables as integers and perform bitwise clear (AND NOT)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/After/BitClear/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

The bits to clear in the state

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/After/BitSet`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Interpret state variables as integers and perform bitwise set (OR)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/After/BitSet/<IDENTIFIER>`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

The bits to set in the state

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/After/Overlay`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Overlay and set the state variables entirely

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/After/Overlay/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

The state value

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Before/BitClear`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Interpret state variables as integers and perform bitwise clear (AND NOT)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Before/BitClear/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

The bits to clear in the state

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Before/BitSet`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Interpret state variables as integers and perform bitwise set (OR)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Before/BitSet/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

The bits to set in the state

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Before/Overlay`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Overlay and set the state variables entirely

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Before/Overlay/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

The state value

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Information`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Realtime command information provided to the acquisition system

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Information/DisplayName`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

The text displayed in the menu entry

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Information/DisplayPriority`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Sort priority for menu entries

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Information/ToolTip`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Detailed tooltip information provided for online help about the command

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Commands/<IDENTIFIER>/Output`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

The command text to write out

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Inputs`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

Input substitution variables

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Inputs/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_command`

Description

The values used for the input

Type

Sequence Selection

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

Cycle Controller

`/<PROFILE>/Components/<IDENTIFIER>/Actions`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The defined cycle actions

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

A single action within the cycle

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting. Additional keys are accepted as part of a higher level composite structure.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`
- `Year`, `Years`, or `y`

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/ClearBypass`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The bypass flags to clear

Type

Flags, a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/ClearContaminate`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The contamination flags to clear

Type

Flags, a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/ClearFlags`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The system flags to clear

Type

Flags, a set of unique text strings

/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Commands**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle

Description

Acquisition command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Description**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle

Description

A human readable description of the action

Type

Text

/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/FlushTime**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle

Description

System data flush time

Type

Real number

Valid range

0.0 < value

Units

s

/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Identifier**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle

Description

An internal identifier of the action

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/SetCut`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The target cut size

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `None` - No size selection
- `PM1, 1,` or `Fine` - Particles are selected to less than 1 µm diameter

- **PM10, 10, or Coarse** - Particles are selected to less than 10 µm diameter
- **PM25, 2.5, 25, or PM2.5** - Particles are selected to less than 2.5 µm diameter

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/SetFlavors`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The system flavors

Type

Flags, a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/BaselineCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Base commands sent to all components for all actions

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/BaselineInstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Base instrument command data for all actions, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/BaselineInstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`<PROFILE>/Components/<IDENTIFIER>/DefaultFlushTime`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The default time to flush the system after executing any action

Type

Real number

Valid range

$0.0 < \text{value}$

Units

s

`<PROFILE>/Components/<IDENTIFIER>/EnableCycle`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Enable the cycle on initialization

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/End/ClearBypass`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The bypass flags to clear

Type

Flags, a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/End/ClearContaminate`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The contamination flags to clear

Type

Flags, a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/End/ClearFlags`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The system flags to clear

Type

[Flags](#), a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/End/Commands`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Acquisition command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/End/Description`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

A human readable description of the action

Type

[Text](#)

`/<PROFILE>/Components/<IDENTIFIER>/End/FlushTime`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

System data flush time

Type

[Real number](#)

Valid range

0.0 < value

Units

s

`/<PROFILE>/Components/<IDENTIFIER>/End/Identifier`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

An internal identifier of the action

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/End/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/End/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/End/SetCut`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The target cut size

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- **None** - No size selection
- **PM1, 1, or Fine** - Particles are selected to less than 1 µm diameter
- **PM10, 10, or Coarse** - Particles are selected to less than 10 µm diameter
- **PM25, 2.5, 25, or PM2.5** - Particles are selected to less than 2.5 µm diameter

`</PROFILE>/Components/<IDENTIFIER>/End/SetFlavors`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The system flavors

Type

Flags, a set of unique text strings

`</PROFILE>/Components/<IDENTIFIER>/InitializeInactive`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Initialize all actions in the cycle as already having been completed

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`</PROFILE>/Components/<IDENTIFIER>/LogActiveAction`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Write logging values reflecting the currently active action

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`</PROFILE>/Components/<IDENTIFIER>/MinimumTime`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The minimum amount of time to run the cycle for, before extending it to the next period

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

`<PROFILE>/Components/<IDENTIFIER>/MinimumTime/Units`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- Millisecond or Milliseconds
- Second, Sec, Seconds, Secs, or s
- Minute, Min, Mins, Minutes, m, or mi
- Hour, Hours, or h
- Day, Days, or d
- Week, Weeks, or w
- Month, Mon, Mons, Months, or mo
- Quarter, Qtr, Qtrs, Quarters, or q
- Year, Years, or y

`<PROFILE>/Components/<IDENTIFIER>/MinimumTime/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/MinimumTime/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Align units in logical time

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

</PROFILE>/Components/<IDENTIFIER>/Period

Required conditions

- **</PROFILE>/Components/<IDENTIFIER>/Name = control_cycle**

Description

The total period of the cycle

Type

Time Interval

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

Undefined values

One hour aligned

</PROFILE>/Components/<IDENTIFIER>/Period/Units

Required conditions

- **</PROFILE>/Components/<IDENTIFIER>/Name = control_cycle**

Description

A logical time unit

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- **Millisecond or Milliseconds**
- **Second, Sec, Seconds, Secs, or s**
- **Minute, Min, Mins, Minutes, m, or mi**
- **Hour, Hours, or h**
- **Day, Days, or d**
- **Week, Weeks, or w**
- **Month, Mon, Mons, Months, or mo**
- **Quarter, Qtr, Qtrs, Quarters, or q**
- **Year, Years, or y**

</PROFILE>/Components/<IDENTIFIER>/Period/Count

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Period/Align`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/ClearBypass`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The bypass flags to clear

Type

Flags, a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/ClearContaminate`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The contamination flags to clear

Type

Flags, a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/ClearFlags`**Required conditions**

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The system flags to clear

Type

[Flags](#), a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Acquisition command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/Description`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

A human readable description of the action

Type

[Text](#)

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/FlushTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

System data flush time

Type

[Real number](#)

Valid range

$0.0 < \text{value}$

Units

s

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/Identifier`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

An internal identifier of the action

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/SetCut`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The target cut size

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `None` - No size selection
- `PM1`, `1`, or `Fine` - Particles are selected to less than 1 μm diameter
- `PM10`, `10`, or `Coarse` - Particles are selected to less than 10 μm diameter
- `PM25`, `2.5`, `25`, or `PM2.5` - Particles are selected to less than 2.5 μm diameter

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Active/SetFlavors`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The system flavors

Type

[Flags](#), a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Duration`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The total time the spin up is active for

Type

[Time Interval](#)

Structure

A time interval consists of a time unit, a number of units to apply, and an alignment setting.

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/Duration/Units`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

A logical time unit

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Millisecond` or `Milliseconds`
- `Second`, `Sec`, `Seconds`, `Secs`, or `s`
- `Minute`, `Min`, `Mins`, `Minutes`, `m`, or `mi`
- `Hour`, `Hours`, or `h`
- `Day`, `Days`, or `d`
- `Week`, `Weeks`, or `w`
- `Month`, `Mon`, `Mons`, `Months`, or `mo`
- `Quarter`, `Qtr`, `Qtrs`, `Quarters`, or `q`

- Year, Years, or y

`<PROFILE>/Components/<IDENTIFIER>/Spinup/Duration/Count`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The number of units to apply in time

Type

Integer

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Spinup/Duration/Align`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Align units in logical time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Spinup/End/ClearBypass`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The bypass flags to clear

Type

Flags, a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Spinup/End/ClearContaminate`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The contamination flags to clear

Type

Flags, a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Spinup/End/ClearFlags`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The system flags to clear

Type

[Flags](#), a set of unique text strings

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/End/Commands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Acquisition command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/End/Description`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

A human readable description of the action

Type

[Text](#)

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/End/FlushTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

System data flush time

Type

[Real number](#)

Valid range

$0.0 < \text{value}$

Units

s

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/End/Identifier`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

An internal identifier of the action

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/End/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/End/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Spinup/End/SetCut`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The target cut size

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `None` - No size selection

- **PM1, 1, or Fine** - Particles are selected to less than 1 µm diameter
- **PM10, 10, or Coarse** - Particles are selected to less than 10 µm diameter
- **PM25, 2.5, 25, or PM2.5** - Particles are selected to less than 2.5 µm diameter

`</PROFILE>/Components/<IDENTIFIER>/Spinup/End/SetFlavors`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

The system flavors

Type

Flags, a set of unique text strings

`</PROFILE>/Components/<IDENTIFIER>/SuspendOnBypass`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_cycle`

Description

Suspend cycle execution on system bypass

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

Realtime Data Export Controller

`</PROFILE>/Components/<IDENTIFIER>/Append`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Append to the existing file instead of overwriting it

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`</PROFILE>/Components/<IDENTIFIER>/BinaryMode`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Open the file in binary mode (no end of line translation)

Type

Boolean, values are **TRUE** or **FALSE** and defaults to **FALSE**

`<PROFILE>/Components/<IDENTIFIER>/Columns`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

The output columns

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Header`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

The header text of the column

Type

Text

Undefined values

Derived from the input

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

The conversion and processing mode used with the column

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `Real` or `Value` - Convert the value to a single real numbered column
- `Time`, `Date`, or `DateTime` - Format the time of the record
- `Flags`, `BitFlags`, `Bits`, or `FlagsBits` - Output the bitwise representation of flags
- `FlagsName`, `FlagsString`, or `StringFlags` - Output the a combination of the internal flag names
- `Script` - Use script code to generate the output value

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/LocalTime`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Time`

Description

Output local system time instead of UTC time

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/TimeFormat`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Time`

Description

Time format string, refer to Qt documentation for available substitutions

Type

Text

Undefined values

ISO8601 compliant

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Bits`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Flags`

Description

Flag name to bit mapping

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Bits/<IDENTIFIER>`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Flags`

Description

Single flag bits

Type

Integer

Valid range

$0 \leq \text{value}$

Undefined values

Not permitted

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Format`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Flags`

Description

The format string used for output

Type

Text

Undefined values

Derived from metadata

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Input`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Flags`

Description

Input data for the column

Type

Sequence Selection

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/MVC`**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Flags`

Description

The missing value code used for output

Type

Text

Undefined values

Derived from metadata

`</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Pad`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Flags`

Description

Output padding character

Type

Text

Undefined values

0

`</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Path`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Flags`

Description

Path within the data to get the value from

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Columns/#0/FlagsRename`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = FlagsName`

Description

Flag code renaming

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`</PROFILE>/Components/<IDENTIFIER>/Columns/#0/FlagsRename/<IDENTIFIER>`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = FlagsName`

Description

Output flag text

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Input`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = FlagsName`

Description

Input data for the column

Type

Sequence Selection

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

`</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Join`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = FlagsName`

Description

The text inserted between flags when combining them

Type

Text

Undefined values

;

`</PROFILE>/Components/<IDENTIFIER>/Columns/#0/MVC`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `</PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = FlagsName`

Description

The missing value code used for output

Type

Text

<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Path**Required conditions**

- <PROFILE>/Components/<IDENTIFIER>/Name = control_export
- <PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = FlagsName

Description

Path within the data to get the value from

Type

Text

<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Code**Required conditions**

- <PROFILE>/Components/<IDENTIFIER>/Name = control_export
- <PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Script

Description

Script code executed to return the output string

Type

Text

<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Input**Required conditions**

- <PROFILE>/Components/<IDENTIFIER>/Name = control_export
- <PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Script

Description

Input data to the script code

Type

Sequence Selection

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Calibration**Required conditions**

- <PROFILE>/Components/<IDENTIFIER>/Name = control_export
- <PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Real

Description

The calibration applied to the input values

Type

Calibration

Structure

A calibration is an array with the coefficients being the array values in ascending power order. All keys are array indices with [real number](#) values. For example, `#0` is the constant offset while `#1` is the slope.

`/<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Format`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `/<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Real`

Description

The format string used for output

Type

Text

Undefined values

Derived from metadata

`/<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Input`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `/<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Real`

Description

Input data for the column

Type

Sequence Selection

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

`/<PROFILE>/Components/<IDENTIFIER>/Columns/#0/MVC`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `/<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Real`

Description

The missing value code used for output

Type

Text

Undefined values

Derived from metadata

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Pad`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Real`

Description

Output padding character

Type

Text

Undefined values

0

`<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Path`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`
- `<PROFILE>/Components/<IDENTIFIER>/Columns/#0/Type = Real`

Description

Path within the data to get the value from

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/Command`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Command line to execute after exporting

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/File`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Output file name

Type

Text

Undefined values

A temporary file

/<PROFILE>/Components/<IDENTIFIER>/Header**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_export

Description

Output a header in the export file

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Join/Quote**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_export

Description

The quote used when the field contains a separator

Type

Text

Undefined values

"

/<PROFILE>/Components/<IDENTIFIER>/Join/QuoteEscape**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_export

Description

The escape used when a quoted field contains a quote

Type

Text

Undefined values

""

/<PROFILE>/Components/<IDENTIFIER>/Join/Separator**Required conditions**

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

The text to join fields with

Type

Text

Undefined values

,

`<PROFILE>/Components/<IDENTIFIER>/Line/Prefix`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

The text prepended to every line

Type

Text

`<PROFILE>/Components/<IDENTIFIER>/Line/Suffix`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

The text appended to every line

Type

Text

Undefined values

A new line character

`<PROFILE>/Components/<IDENTIFIER>/LoadArchived`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Load archived data for export on initialization

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/NoFile`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Disable file output (used for interface writing)

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`</PROFILE>/Components/<IDENTIFIER>/Processing/Additional`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Additional inputs requested for processing

Type

Sequence Selection

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

`</PROFILE>/Components/<IDENTIFIER>/Processing/Components`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Processing chain stages

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices.

`</PROFILE>/Components/<IDENTIFIER>/Processing/Components/#0/Name`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Data generating component name

Type

Text

`</PROFILE>/Components/<IDENTIFIER>/Processing/Input/End`**Required conditions**

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

The end time to invoke the generator with

Type

Real number

Units

s

Undefined values

Not permitted

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/End`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

The end time to invoke the generator with

Structure

This is an absolute point in time specified as the number of seconds from the Unix epoch (1970-01-01T00:00:00Z).

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/File`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Input file name used with the generator

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/IgnoreTimes`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Do not provide times to the input generator

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Merge`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_export`

Description

Allow merging of identical processing inputs

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Name**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_export

Description

Data generating component name

Type

Text

/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Start**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_export

Description

The start time to invoke the generator with

Type

Real number

Units

s

Undefined values

Not permitted

/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Start**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_export

Description

The start time to invoke the generator with

Structure

This is an absolute point in time specified as the number of seconds from the Unix epoch (1970-01-01T00:00:00Z).

/<PROFILE>/Components/<IDENTIFIER>/Processing/Input/Station**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_export

Description

Input station used with the generator

Type

Text

/<PROFILE>/Components/<IDENTIFIER>/Processing/Merge**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_export

Description

Allow merging of identical processing chains

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

/<PROFILE>/Components/<IDENTIFIER>/Processing/MergeKey**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_export

Description

The unique identifier that must match for chain merging

Type

Text

Generic Trigger Condition Controller

/<PROFILE>/Components/<IDENTIFIER>/Actions**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger

Description

The defined trigger actions

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, #0 is the first element of the array. The paths represented here use #0 as a placeholder for all indices.

/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/ClearBypass**Required conditions**

- /<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger

Description

The bypass flags to clear

Type

Flags, a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/ClearContaminate`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

The contamination flags to clear

Type

`Flags`, a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/ClearFlags`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

The system flags to clear

Type

`Flags`, a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Commands`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Acquisition command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Continuous`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Continually activate the event as long as the trigger condition is met

Type

`Boolean`, values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Dependencies/Acquire`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Identifier tags acquired on action execution

Type

Flags, a set of unique text strings

</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Dependencies/Hold**Required conditions**

- </PROFILE>/Components/<IDENTIFIER>/Name = control_trigger

Description

The length of time the tags altered by the action are held in limbo after the action executes

Type

Real number

Valid range

0.0 < value

Units

s

Undefined values

Not permitted

</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Dependencies/Release**Required conditions**

- </PROFILE>/Components/<IDENTIFIER>/Name = control_trigger

Description

Identifier tags released on action execution

Type

Flags, a set of unique text strings

</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Description**Required conditions**

- </PROFILE>/Components/<IDENTIFIER>/Name = control_trigger

Description

A human readable description of the action

Type

Text

</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Event**Required conditions**

- </PROFILE>/Components/<IDENTIFIER>/Name = control_trigger

Description

Event text to generate

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/EventRealtime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Flag the generated event as realtime, so it appears on the message display

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/FlushTime`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

System data flush time

Type

Real number

Valid range

0.0 < value

Units

s

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Identifier`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

An internal identifier of the action

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/InstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Instrument command data, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/InstrumentCommands/<IDENTIFIER>`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/SetCut`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

The target cut size

Type

[Enumeration](#)

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- `None` - No size selection
- `PM1`, `1`, or `Fine` - Particles are selected to less than 1 μm diameter
- `PM10`, `10`, or `Coarse` - Particles are selected to less than 10 μm diameter
- `PM25`, `2.5`, `25`, or `PM2.5` - Particles are selected to less than 2.5 μm diameter

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/SetFlavors`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

The system flavors

Type

[Flags](#), a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

The triggers for the action

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Exclude`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Identifier tags that prevent trigger execution

Type

[Flags](#), a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Invert`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Invert the trigger and activate when the condition if not met

Type

[Boolean](#), values are `TRUE` or `FALSE` and defaults to `FALSE`

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Require`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Identifier tags required for the trigger to execute

Type

[Flags](#), a set of unique text strings

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

The type of the condition being checked

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of possible choices. Possible values:

- **Threshold** or **Limit** - Trigger when the input meets a threshold relative to a constant value
- **AND** or **Composite** - Combine multiple triggers and require them all to be met
- **OR** or **Any** - Combine multiple triggers and require any of them to be met
- **Always** - Always trigger
- **Never** - Never trigger
- **Flags** or **Flag** - Trigger when a set of system flags are present
- **Spike** or **Risk** - Trigger when a spike is detected in a smoothed value
- **Stable** or **Smoother** - Trigger when a smoothed value achieves stability
- **Range** - Trigger when the input is within a range
- **Angle**, **Wind**, or **Winds** - Trigger when the input angle is within a range (wrapping around 360)
- **Script** - Use script code to check a condition

`</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Triggers`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = AND`

Description

The triggers to combine together, requiring them all to be met

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Triggers/#0`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = AND`

Description

A single trigger

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Triggers`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = OR`

Description

The triggers to combine together, requiring them any of them to be met

Keys

The keys immediately below this are array indices. These are indexes into an ordered list, with the first element being zero. For example, `#0` is the first element of the array. The paths represented here use `#0` as a placeholder for all indices.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Triggers/#0`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = OR`

Description

A single trigger

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Polarity`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Always`

Description

The output trigger result

Type

Boolean, values are `TRUE` or `FALSE` and defaults to `FALSE`

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Polarity`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Never`

Description

The output trigger result

Type

[Boolean](#), values are [TRUE](#) or [FALSE](#) and defaults to [FALSE](#)

[/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Flags](#)

Required conditions

- [/<PROFILE>/Components/<IDENTIFIER>/Name](#) = control_trigger
- [/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type](#) = Flags

Description

The required flags

Type

[Flags](#), a set of unique text strings

[/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Input](#)

Required conditions

- [/<PROFILE>/Components/<IDENTIFIER>/Name](#) = control_trigger
- [/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type](#) = Flags

Description

Input to check for flags

Type

[Sequence Selection](#)

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

[/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Input](#)

Required conditions

- [/<PROFILE>/Components/<IDENTIFIER>/Name](#) = control_trigger
- [/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type](#) = Spike

Description

The input value to smooth

Type

[Dynamic Real Value Input](#)

Structure

This is a specification of a single real valued input used in calculation. The specification generally takes the form of either a value from the data or a constant.

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Smoother`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Spike`

Description

The smoother used for spike detection

Type

[Baseline Smoother](#)

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the `Type enumeration` key. The smoother is used for spike detection. This means that the type of smoother selected should provide an indicator of when the incoming value is considered a spike.

Undefined values

A 30 minute long fixed time smoother, with a threshold of -4.0 (four times greater than the smoothed value)

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Input`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Stable`

Description

The input value to smooth

Type

[Dynamic Real Value Input](#)

Structure

This is a specification of a single real valued input used in calculation. The specification generally takes the form of either a value from the data or a constant.

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Smoother`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Stable`

Description

The smoother used for stability detection

Type

Structure

A baseline smoother is a construct that is used to generate a smoothed output based on an input value sequence. The possible valid configuration parameters are determined by the [Type enumeration](#) key. The smoother is used for stability detection. This means that the type of smoother selected should provide an indicator of when the smoothed value is stable.

Undefined values

A 90 second long smoother with a required RSD of less than 0.001

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Input`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Range`

Description

The input value check

Type

[Dynamic Real Value Input](#)

Structure

This is a specification of a single real valued input used in calculation. The specification generally takes the form of either a value from the data or a constant.

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Maximum`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Range`

Description

The maximum value to trigger on

Type

[Real number](#)

`<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Minimum`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Range`

Description

The minimum value to trigger on

Type

Real number

`</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/End`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Angle`

Description

The end angle to trigger, if the end is less than the start, the range wraps around through 360

Type

Real number

`</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Input`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Angle`

Description

The input angle check

Type

Dynamic Real Value Input

Structure

This is a specification of a single real valued input used in calculation. The specification generally takes the form of either a value from the data or a constant.

`</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Start`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Angle`

Description

The start angle to trigger, if the end is less than the start, the range wraps around through 360

Type

Real number

`</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Code`

Required conditions

- `</PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `</PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Script`

Description

Script code executed to return the the boolean state of the trigger

Type

Text

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Input`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Script`

Description

Input to the script code

Type

Sequence Selection

Structure

A sequence selection is a means of selecting a sequence of values in the data. The specification consists of selections for the various components of the sequence name.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Input`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Threshold`

Description

The input angle check

Type

Dynamic Real Value Input

Structure

This is a specification of a single real valued input used in calculation. The specification generally takes the form of either a value from the data or a constant.

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Operation`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`
- `/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type = Threshold`

Description

The comparison operation performed

Type

Enumeration

Values

This is an enumeration value. It accepts a single string (not case sensitive) from a list of

possible choices. Possible values:

- `GreaterThan` or `Greater` - Trigger when the input is strictly greater than the threshold
- `GreaterThanOrEqual`, `GreaterEqual`, or `GreaterThanOrEqualTo` - Trigger when the input is greater than or equal to the threshold
- `LessThan` or `Less` - Trigger when the input is strictly less than the threshold
- `LessThanOrEqual`, `LessEqual`, or `LessThanOrEqualTo` - Trigger when the input is less than or equal to the threshold
- `Equal` or `Exactly` - Trigger when the input is exactly equal to the threshold
- `NotEqual` - Trigger when the input is not equal to the threshold

`/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Threshold`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name` = control_trigger
- `/<PROFILE>/Components/<IDENTIFIER>/Actions/#0/Triggers/#0/Type` = Threshold

Description

The threshold value to compare against

Type

Real number

`/<PROFILE>/Components/<IDENTIFIER>/BaselineCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name` = control_trigger

Description

Base commands sent to all components for all actions

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`/<PROFILE>/Components/<IDENTIFIER>/BaselineInstrumentCommands`

Required conditions

- `/<PROFILE>/Components/<IDENTIFIER>/Name` = control_trigger

Description

Base instrument command data for all actions, keyed by the instrument identifier (e.x. X1)

Keys

The keys immediately below this are unique names selected by the user. These names can be any non-empty text and are generally used to designate different components to be configured. In the paths listed `<IDENTIFIER>` is used as a placeholder to real name selected during configuration.

`<PROFILE>/Components/<IDENTIFIER>/BaselineInstrumentCommands/<IDENTIFIER>`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Single instrument command data

Keys

The keys below this one have an arbitrary structure. That is, the exact contents are not directly interpreted by the system so they are defined by conventions or subsequent usage.

`<PROFILE>/Components/<IDENTIFIER>/DefaultFlushTime`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

The default time to flush the system after executing any action

Type

Real number

Valid range

0.0 < value

Units

s

`<PROFILE>/Components/<IDENTIFIER>/EnableTrigger`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Enable the all triggers on initialization

Type

Boolean, values are TRUE or FALSE and defaults to FALSE

`<PROFILE>/Components/<IDENTIFIER>/LogActiveAction`

Required conditions

- `<PROFILE>/Components/<IDENTIFIER>/Name = control_trigger`

Description

Write logging values reflecting the currently active action

Type

Boolean, values are TRUE or FALSE and defaults to FALSE