CPD3 Command Line Interface

Table of Contents

Quick Reference

Access and Export

- da.get - Read data from the archive
- da.avg - Average data
- da.export - Simple data export
- da.generate.edited - Generate edited data
- da.indirect - Indirect value lookup
- da.output.ebas - Generate EBAS data files
- da.output.netcdf - Generate NetCDF data files
- da.pass - Flag data as having passed mentor QA/QC
- da.select - Select data stream components
- da.summary.general - General data summary
- da.summary.timeline - Timeline data summary
- da.where - Conditionally select dat
- da.multiplex - Combine multiple data streams

Correction and Calculation

- da.calc.dewpoint - Dewpoint, temperature and RH calculation
- da.calc.integratesize - Calculate size distribution integrated properties
- da.calc.intensives - Calculate optical intensive properties
- da.calc.lwcprobe - Liquid water content probe calculation
- da.calc.opticaldepth - Calculate integrated optical depths
- da.calc.pitotflow - Calculate flow from a pitot tube sensor
- da.calc.purpleair - PurpleAir derived parameter calculation
- da.calc.radiation - Surface radiation calculation
- da.calc.resistancetemperature - Calculate temperature from resistance
- da.calc.simple - Simple output calculation
- da.calc.solarposition - Calculate solar position parameters
- da.corr.andersonogren1998 - Apply Anderson and Ogren 1998 truncation correction
- da.corr.bond1999 - Apply the Bond et al. 1999 correction
- da.corr.dilution - Correct data for dilution
- da.corr.mueller2011 - Apply Müller et al (2011) truncation correction
- da.corr.mueller2014 - Apply the Müller 2014 (aka CTS) correction

- da.corr.removecontam - Remove contaminated values
- da.corr.removeundefined - Remove data that is undefined
- da.corr.script - General purpose script processing
- da.corr.stp - Correction to STP
- da.corr.virkkula2005 - Apply the Virkkula et al. 2005 correction with 2010 errata
- da.corr.wavelength - Adjust data wavelengths
- da.corr.weiss - Weiss loading correction
- da.smooth.1plp - Apply a single pole low pass digital filter
- da.smooth.3rssh - Apply Tukey 3RSSH smoothing
- da.smooth.4plp - Apply a four pole low pass digital filter
- da.smooth.df - Apply a manually specified digital filter
- da.smooth.fourier - Apply Fourier transformed frequency space based smoother
- da.smooth.segment - Average segments of data

Plotting

- da.plot.allan - Allan Plot
- da.plot.cdf - Cumulative Distribution
- da.plot.cycle - Cycle plot
- da.plot.density - Density plot
- da.plot.layout - Pre-defined layout
- da.plot.pdf - Probability Distribution
- da.plot.scatter - Scatter plot
- da.plot.timeseries - Time series plot
- da.show.allan - Allan Plot
- da.show.cdf - Cumulative Distribution
- da.show.cycle - Cycle plot
- da.show.density - Density plot
- da.show.layout - Pre-defined layout
- da.show.pdf - Probability Distribution
- da.show.scatter - Scatter plot
- da.show.timeseries - Time series plot

External Data Conversion

- da.acquire.2b.ozone205 - Acquire 2B Ozone Monitor model 205 data

- da.acquire.ad.cpcmagic200 - Acquire Aerosol Dynamics Magic 200 CPC data
- da.acquire.aerodyne.caps - Acquire Aerodyne CAPS data
- da.acquire.bmi.cpc1710 - Acquire BMI MCPC 1710 data
- da.acquire.bmi.cpc1720 - Acquire BMI MCPC 1720 data
- da.acquire.campbell.cr1000gmd - Acquire data from a Campbell CR1000 with GMD Aerosols firmware
- da.acquire.csd.pops - Acquire CSD POPS data
- da.acquire.dmt.bcp - Acquire DMT BCP data
- da.acquire.dmt.ccn - Acquire DMT CCN data
- da.acquire.dmt.pax - Acquire DMT PAX data
- da.acquire.eigenbrodt.nmo191 - Acquire Eigenbrodt NMO191
- da.acquire.ecotech.nephaurora - Acquire Ecotech Aurora 3000 and 4000 Nephelometer data
- da.acquire.generic.metar - Acquire METAR data
- da.acquire.gmd.clap3w - Acquire GMD CLAP-3W or BMI TAP data
- da.acquire.gmd.cpcpulse - Acquire GMD pulse counter CPC data
- da.acquire.grimm.opc110x - Acquire Grimm 110x data
- da.acquire.magee.aethalometer162131 - Acquire Magee Aethalometer AE16, AE21, or AE31 data
- da.acquire.magee.aethalometer33 - Acquire Magee Aethalometer AE33 data
- da.acquire.magee.tca08 - Acquire Magee TCA 08 data
- da.acquire.maycomm.tdl - Acquire Maycomm TDL data
- da.acquire.pms.lasair - Acquire PMS LASAIR OPC data
- da.acquire.purpleair.pa2 - Acquire PurpleAir PA-II JSON data
- da.acquire.rmy.wind86xxx - Acquire RM Young 86000 series wind data
- da.acquire.rr.neph903 - Acquire Radiance Research M903 Nephelometer data
- da.acquire.rr.psap1w - Acquire Radiance Research single wavelength PSAP data
- da.acquire.rr.psap3w - Acquire Radiance Research three wavelength PSAP data
- da.acquire.thermo.maap5012 - Acquire Thermo MAAP 5012 data
- da.acquire.tsi.cpc3010 - Acquire TSI 3010 CPC data
- da.acquire.tsi.cpc302x - Acquire TSI 302X CPC data
- da.acquire.tsi.cpc377x - Acquire TSI 377X CPC data
- da.acquire.tsi.cpc3781 - Acquire TSI 3781 CPC data
- da.acquire.tsi.cpc3783 - Acquire TSI 3783 CPC data
- da.acquire.tsi.mfm4xxx - Acquire TSI MFM 4xxx data
- da.acquire.tsi.neph3563 - Acquire TSI 3563 Integrating Nephelometer data
- da.acquire.vaisala.pwdx2 - Acquire Vaisala PWD12, PWD22, or PWD52 data

- da.acquire.vaisala.wmt700 - Acquire Vaisala WMT700 data
- da.acquire.vaisala.wxt5xx - Acquire Vaisala WXT5xx data
- da.generate.fromebas - Generate data from the EBAS/WDCA archive
- da.import.cpd2 - CPD2 Data Import
- da.import.ebas - EBAS Data Import

Internal System Programs

- da.archive - Write data to the archive
- da.email.send - Send a periodic processing report
- da.maintenance.archive - Perform periodic archive maintenance tasks
- da.output.sqldb - Write data to an external SQL database
- da.output.upload - Upload files to external targets
- da.process.authorize - Authorize certificates to process incoming data
- da.process.incoming - Process incoming data into the local archive
- da.process.unpack - Unpack data transfer files
- da.smooth.realtime - Apply realtime averaging to data
- da.sync.authorize - Authorize certificates to synchronize data
- da.sync.client - Synchronize with a remote server
- da.sync.connect - Manually synchronize with a remote server
- da.sync.files - Synchronize via transferred files
- da.tap - Tap the data stream
- da.tasks - Run all periodic tasks
- da.time.limit - Limit data time
- da.time.shift - Shift data time
- da.transfer.upload - Upload new data to a remote server
- da.update.passed - Update passed data
- da.update.plots - Update plots
- da.config.text - Display or edit flat text configuration
- da.output.cpd2 - Generate CPD2 style records
- da.sync.peer - Manual tunnel data synchronization
- da.sync.read - Read a synchronization stream
- da.sync.write - Write a synchronization stream

Access and Export

These are the high level programs used to access data and produce externally readable representations of it.

da.get

This component reads data directly from the main archive. It is normally used as the start of a data extraction and processing chain. Data generated by it are not in a human readable format. To display the data use a component like da.export instead.

Usage

da.get [station] variables times [archive]

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

Examples

General usage

This retrieves the record **S11a** (which is an alias for several relevant nephelometer parameters) for the station **brw** from 2010-01-01 to 2010-01-10 from the **raw** archive.

```
da.get brw S11a 2010-01-01 2010-01-10
```

Specifying a list of variables

This retrieves the given variables from the **clean** archive and also demonstrates more relaxed time specification.

```
da.get brw BsB_S11,BsG_S11,BsR_S11 2010 5 2010 7 clean
```

Advanced variable selection

This retrieves only variable **BsB_S11** from the **raw** archive of the implied station (from the working directory or other information) but only the **PM1** cut size data. It also retrieves the variable from the **spo clean** archive with both cut sizes. The data are retrieved from 2010-01-01 to 2010-01-02.

```
da.get ::BsB_S11:pm1 spo:clean:BsB_S11 2010:1
```

da.avg

This component applies conventional fixed length averaging to input data. Normally it also removes contaminated values from the averages. Note that some types of averaging require inputs beyond the values being averaged. For example, to average raw absorptions, transmittance and path length are required because the average is performed with a difference measurement.

Usage

da.avg [switches...] [[station] variables times [archive]|[file]]

Switches

--contamination=**STRING**

This defines the contamination mode name in effect. The contamination mode determines which variables are removed when they are flagged as contaminated. This often corisponds with various profiles for other station specific configuration. For example the "aerosol" mode generally removes scattering, absorption, extinction and concentrations. The special mode "none" or "disable" turns off all contamination removal and values are averaged regardless of their contaminated state.

Default: **aerosol**

--continuous[=BOOLEAN]

If this option is set then the averages produced are the maximum continuous averages within the binning interval. This allows difference measurements to be repeatable but may cause splitting within the interval.

Default: **Disabled**

--cover=NUMBERS

This is the fraction of data required to exist for an average bin to be produced at all. For example if this is set to 0.95 then averages will only be produced when the fraction of missing data in a bin is greater than 95% of the total bin. So, for a one hour bin that would require at least 57 minutes of valid data. Setting this to undefined disables the requirement.

Default: **Disabled**

--gap=INTERVAL

If two values are seperated by this much time then the average is split. Even if there is no gap the values may still be split into separate bins if the interval demands it. An undefined gap allows for infinite separation.

Default: **Infinite**

Undefined intervals are accepted. The interval must be greater than or equal to zero length.

--interval=INTERVAL

This is width of the bin to average on. All data within each interval is averaged together to produce the output. An undefined interval averages all available data together.

Default: **One hour**

Undefined intervals are accepted. The interval must be greater than zero in length.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of

selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will average input data to one hour intervals and remove contamination from the standard set of variables involved in an aerosol system (generally scattering, absorption, extinction and concentrations).

```
da.avg bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.avg input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.avg
```

Interval and coverage requirement

This will average data to one day interval requiring at least 95% of data within each day to exist (1368 minutes). Contaminated data is not excluded from the averages.

```
da.avg --contamination=none --cover=0.95 --interval=1da bnd S11a 2015-05-01 2015-
05-03
```

Infinite continuous averages

This will average all input data to the maximum extent possible but will break up those averages whenever there is a discontinuity in the input data. This also excludes contaminated values for all standard aerosol parameters.

```
da.avg --continuous --interval= bnd S11a 2015-05-01 2015-05-03
```

da.export

This component provides a simple mechanism to write output data in CSV-like formats. This can include "true" CSV data as well as variations like CPD2 style and tab separated forms.

Usage

da.export [switches...] [[station] variables times [archive]|[file]]

Switches

--mode=TYPE

This is the base export mode. It defines the defaults for all other options. Those defaults can be overridden by setting the option directly. For some export modes (XML and raw) there are no other options so any specified are ignored. Those exports are direct translations of the input data stream.

Default: **MS Excel compatible**

The possible values are:

--mode=CPD1

CPD1 compatible (CSV, station, year and doy)

--mode=CSV

General CSV (CSV, year-doy, blank-mvc)

--mode=Excel

Same as --mode=**xl**

--mode=IDL

IDL compatible (space-separated, Julian day, standard mvc)

--mode=IDLLegacy

IDL legacy code compatible (station, year and doy)

--mode=JSON

JSON data (no other options applied)

--mode=R

R/S compatible (CSV, epoch time, "NA"-mvc)

--mode=S

Same as --mode=**R**

--mode=XML

XML data (no other options applied)

--mode=direct1

Direct base64 data in version 1 format (no other options applied)

--mode=raw

Raw binary data (no other options applied)

--mode=raw1

Raw binary data in version 1 format (no other options applied)

--mode=xl

MS Excel compatible (CSV, date-time, blank-mvc)

--bounds[=BOOLEAN]

If set then columns for containing the absolute minimum and maximum of the data in the average will be output.

--count[=BOOLEAN]

If enabled then columns with the number of observations for each column will be included.

Default: **Enabled**

--cover[=BOOLEAN]

If set then columns representing the fraction of data available in each average will be output.

--cut-split[=BOOLEAN]

If set then variables will be renamed/split based on their cut size even if they do not overlap. If set to false then automatic detection is disabled and data are never split. Disabling this can have unpredictable results if used with data that does overlap.

Default: **Automatic**

--cut-string[=BOOLEAN]

If set then the output includes an indicator containing a semicolon delimited list of cut sizes.

--end[=BOOLEAN]

If set then columns representing the value at the end of the average will be output.

--fill=INTERVAL

This is maximum interval to create filling lines at. For example, five minutes would create an output every five minutes during gaps, starting at the beginning of the gap. Setting it to undefined creates a single output for the entire gap. This is best coupled with averaging of the same interval for uniform data output.

Undefined intervals are accepted. The interval must be greater than zero in length.

--flags=TYPE

Sets the flags breakdown and output mode. This determines how flags values are written out.

The possible values are:

--flags=0x

Output flags out as hexadecimal numbers with a leading "0x".

--flags=breakdown

Output flags as a series of zero or one values corresponding to each possible flag.

--flags=default

Output flags out like a conventional value.

--flags=list

Output flags as a list of flag names in the field.

--format=STRING

If set then this is the format (like printf(2) or of the form "000.000") used to output all numeric values.

--header-cut[=BOOLEAN]

If set then a line with a cut size for the whole column is included in the header.

--header-description[=BOOLEAN]

If set then a line with field descriptions is included in the header.

--header-flags[=BOOLEAN]

If set then lines for each possible flag will be included in the header.

--header-mark=STRING

If set then this is added to the start of all header lines.

--header-mvcs[=BOOLEAN]

If set then a line with field missing value codes is included in the header.

--header-names[=BOOLEAN]

If set then a line with field names is included in the header.

--header-names-stderr[=BOOLEAN]

If set then a line with field names is output to standard error (regardless of invocation context).

--header-wavelength[=BOOLEAN]

If set then a line with field wavelengths (including possibly time dependent information) is included in the header.

--join=TYPE

This is the base field joining mode. The mode set here determines the defaults used to join together the output. Those options can be overridden individually as well.

The possible values are:

--join=csv

Standard comma separated values. Fields are joined together with a comma (ASCII 0x2C) and quoted with a double quote (ASCII 0x22).

--join=space

Space separated values. Fields are joined together with a space (ASCII 0x20) and quoted with double quotes (ASCII 0x22).

--join=tab

Tab separated values. Fields are joined together with a tab (ASCII 0x09).

--join-delimiter=STRING

If set then this overrides the string used to delimit the join between multiple fields.

--join-quote=STRING

If set then this overrides the string added to the start and end of a field if it needs to be quoted.

--join-quote-escape=STRING

If set then this overrides the string used to replace all instances of the quote string within a quoted field.

Default: **The quote string repeated three times**

--mean[=BOOLEAN]

If set then the primary data columns will contain the unweighted means when they are available instead of coverage weighted and/or difference measurements.

--mvc=TYPE

This is the MVC output style. Missing or undefined values are written in accordance with this style. Cannot be used with a direct specification of the MVC.

The possible values are:

--mvc=NA

Output the string "NA" for missing values.

--mvc=blank

Output an empty field for missing values.

--mvc=default

Use the default MVC for each value.

--mvc-all=STRING

If set then this is the MVC string used for all values. Cannot be used with a MVC mode specification.

--mvc-flag=TYPE

Sets the missing value flag mode. If enabled missing value flags are fields that are either zero or one to indicate if the value corresponding to the flag was defined in the line.

The possible values are:

--mvc-flag=disable

Disable MVC flags.

--mvc-flag=end

Output MVC flags at the end of the record in the same order as the prior fields in the record.

--mvc-flag=follow

Output MVC flags immediately following the value they correspond to.

--numeric-only[=BOOLEAN]

If set then only values with a numeric representation will be output, all others will be considered missing.

--paths[=BOOLEAN]

If set then all complex values have their component paths output separately, resulting in one column for each path.

--quantiles[=BOOLEAN]

If set then columns for the standard quantiles (0.00135, 0.00621, 0.02275, 0.05, 0.06681, 0.15866,

0.25, 0.30854, 0.5, 0.69146, 0.75, 0.84134, 0.93319, 0.95, 0.97725, 0.99379, and 0.99865) of the data in the average will be output.

--squash=**INTERVAL**

This sets a limit below which data are combined into a single record. That is, if two variables have values separated by less than this threshold, they will be combined into a single output record.

Undefined intervals are accepted. The interval must be greater than zero in length.

--station[=**BOOLEAN]**

If set then a station code is included in the output.

--stddev[=**BOOLEAN]**

If enabled then columns for the standard deviation of the averages will be output.

Default: **Enabled**

--time-doy[=**BOOLEAN]**

If set then the output includes the fractional day of year (with January 1 = 1.00000).

--time-epoch[=**BOOLEAN]**

If set then the output includes the number of seconds since 1970-01-01T00:00:00Z.

--time-excel[=**BOOLEAN]**

If set then the output includes the date and time in a format recognized by Excel.

--time-fractional-year[=**BOOLEAN]**

If set then the output includes the fractional year with enough precision to represent seconds.

--time-iso[=**BOOLEAN]**

If set then the output includes a date-time formatted in compliance with ISO 8601.

--time-julian-day[=**BOOLEAN]**

If set then the output includes the fractional Julian day. That is the fractional number of days since 12:00 January 1, 4713 BC.

--time-yeardoy[=**BOOLEAN]**

If set then the output includes the year and fractional day of year (with January 1 = 1.00000) as separate fields.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular

expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Default MS Excel formatted

```
da.export bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.export input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.export
```

CSV with year and DOY

```
da.export --mode=CSV bnd S11a 2015-05-01 2015-05-03
```

Excel with station and fractional year

```
da.export --mode=xl --station --time-fractional-year bnd S11a 2015-05-01 2015-05-03
```

da.generate.edited

This component generates edited data for a station. Edited data has standard corrections and mentor edits applied to it to form "final" output data. The output data normally contains both the high time resolution data (normally output as the "clean" archive) as well as the averaged data (normally output as the "avgH" archive).

Usage

da.generate.edited [switches...] [station] times

Switches

--profile=STRING

This is the profile name to generate data for. Different editing profiles specify different sets of data or methods of generating final data within the same station. Consult the station configuration for available profiles.

Default: **aerosol**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

Examples

Generate edited data

This will generate edited data for the "aerosol" profile.

```
da.generate.edited
```

da.indirect

This component provides a mechanism to perform indirect value lookups from one value into another.

Usage

da.indirect [switches...] [[station] variables times [archive]|[file]]

Switches

--from=**SELECTION**

This is the input that specifies the origin or master of the lookup.

--into=**SELECTION**

This is the value that the indirect lookup is performed into.

--output=**SELECTION**

This is the output that the lookup generates.

--path=**STRINGS**

This is the path within the indirect to look up the value at.

--reference-path=**STRINGS**

This is the path within the into origin that is selected for substitution.

--string=**STRINGS**

If not empty, then this string is used as the output value, after substitutions have been applied.

--target-path=**STRINGS**

This is the path within the output that the result is placed into.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

da.output.ebas

This generates files in the for accepted by the EBAS/WDCA archive. These files are derived from the NASA/Ames format with customized extensions specific to EBAS. The output is one or more specially named files in an output directory.

Usage

da.output.ebas [switches...] station... times

Switches

--directory=**DIRECTORY**

This is the directory files will be placed in.

Default: **Current directory**

--output=**STRING**

This is the output configuration to generate. Output configurations generally corispond to EBAS data levels. Consult the profile configuration for available outputs. This is a regular expression that can match multiple outputs.

Default: **All available**

--profile=**STRING**

This is the profile name to output. Multiple profiles can be defined to specify different sets of data that can be output independently. Consult the station configuration for available profiles.

Default: **aerosol**

--quiet[=**BOOLEAN]**

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--type=**STRING**

This is the type of output to generate. Different output types generally correspond to different cut size and instrument pairs. Consult the profile configuration for available types. This is a regular expression that can match multiple types.

Default: **All available**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of

selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

Examples

Output EBAS data

This will output data for the default "aerosol" profile.

```
da.output.ebas bnd 2015-05-01 2015-05-03
```

da.output.netcdf

This generates files formatted with NetCDF4. The specific output definitions are determined by the active configuration. Normally this is used to generate files suitable for submission to NCEI. The output is one or more specially named files in an output directory.

Usage

da.output.netcdf [switches...] station... times

Switches

--directory=**DIRECTORY**

This is the directory files will be placed in.

Default: **Current directory**

--profile=**STRING**

This is the profile name to output. Multiple profiles can be defined to specify different sets of data that can be output independently. Consult the station configuration for available profiles.

Default: **aerosol**

--quiet[=**BOOLEAN]**

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

Examples

Output NetCDF4 data

This will output data for the default "aerosol" profile.

```
da.output.netcdf bnd 2015-05-01 2015-05-03
```

da.pass

This component flags as having been examined by the station mentor and being ready for final export to externally visible locations. It normally also updates the internal clean and averaged data source with the newly passed data.

Usage

da.pass [switches...] station... [times]

Switches

--comment=STRING

This is an optional comment to attach to the pass operation. Ideally this should explain the reasoning for the (re)passing if there is anything anomalous about it.

--detached[=BOOLEAN]

This option causes the data update to be executed in detached processes if available. This allows for better isolation in the event of failure or other problems.

Default: **Enabled for more than 31 days**

--noupdate[=BOOLEAN]

When enabled, this causes the immediate data update to be skipped. The result is that the clean and averaged data are not updated until the next time the station tasks are run at the appropriate level. This normally means that the data will not be updated until the following night when the execution tasks run automatically.

--profile=STRING

This is the profile name to pass. Multiple profiles can be defined to specify different sets of data that can be passed independently. Consult the station editing configuration for available profiles.

Default: **aerosol**

--quiet[=BOOLEAN]

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

Examples

Pass data

This will pass data for the default "aerosol" profile.

```
da.pass bnd 2015-05-01 2015-05-03
```

da.select

This component provides utilities to select specific components out of the whole data stream. This includes including or excluding specific variables only and selecting paths from within complex types.

Usage

da.select [switches...] [[station] variables times [archive]|[file]]

Switches

--capture-archive=**STRINGS**

This is a regular expression the archive must match to be changed. When set the captures ${1}, ${2}, ... are available for substitution in the output archive.

--capture-station=**STRINGS**

This is a regular expression the station must match to be changed. When set the captures ${1}, ${2}, ... are available for substitution in the output station.

--capture-variable=**STRINGS**

This is a regular expression the variable must match to be changed. When set the captures ${1}, ${2}, ... are available for substitution in the output variable.

--end=**TIME**

This is the end time to output data. If this is set then no data after this time is output.

--exclude=**SELECTION**

This option sets the values that are excluded from the output.

Default: **Nothing**

--explicit-meta[=**BOOLEANS]**

When enabled this changes the handling of metadata so it must be matched manually. Normally metadata is unchanged by path access and is passed if the corresponding normal value passes.

--include=**SELECTION**

This option sets the values that are included in the output.

Default: **Everything**

--path=**STRINGS**

This is the path applied to values. This only makes sense for values with child types (e.x. hashes or metadata). The value accessed at the path will become the new root for everything it is applied to.

Default: **None**

--path-apply=**SELECTION**

This option sets the values that have the path applied to them in the output.

Default: **Everything**

--path-remove[=**BOOLEANS]**

When enabled this changes the path to be removed from the value instead of applying it and passing the value. This is, this causes the path within values to be deleted.

--remove-undefined[=**BOOLEANS]**

When enabled this this removes all undefined values from the output.

--set-apply=SELECTION

This option sets the values that have the alteration of station, archive and/or variable applied to them.

Default: **Everything**

--set-archive=STRINGS

This is the archive that values are set to.

Default: **No change**

--set-duplicate[=BOOLEANS]

When enabled this causes a copy of the value to be generated before the station, archive, or variable are changed.

--set-flavor-add=STRINGS

This is a flavor that is added.

Default: **No change**

--set-flavor-remove=STRINGS

This is a flavor that is removed.

Default: **No change**

--set-station=STRINGS

This is the station that values are set to.

Default: **No change**

--set-variable=STRINGS

This is the variable that values are set to.

Default: **No change**

--start=TIME

This is the start time to output data. If this is set then no data before this time is output.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Select variable

This creates output that only contains a single variable.

```
da.select --include=bnd:raw:BsG_S11 bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.select --include=bnd:raw:BsG_S11 input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.select --include=bnd:raw:BsG_S11
```

Extract wavelength

This creates output that contains only the wavelength information from a single variable.

```
da.select --explicit-meta --include=bnd:raw_meta:BsG_S11 --path=^Wavelength
--remove-undefined --set-archive=raw bnd S11a 2015-05-01 2015-05-03
```

da.summary.general

This component provides a means of getting a high level simple summary of input data. This involves things like the latest or current wavelengths of variables, their descriptions, the originating instrument similar parameters.

Usage

da.summary.general [switches...] [[station] variables times [archive]|[file]]

Switches

--mode=TYPE

This is the base summary mode. This determines the results output as well as the method of displaying them.

Default: **Per-variable summary**

The possible values are:

--mode=instrument

Summary output on an instrument

--mode=variable

Summary output on a per-variable basis

--cover[=BOOLEAN**]**

If set then the summary includes the global coverage percentage of each output.

--data=STRING...

This sets any additional data paths to include in the final summary output.

--description[=BOOLEAN**]**

If set then the summary includes the variable description from the metadata.

--display=TYPE

This is the mode used to format and display the final results.

Default: **Determined by primary mode**

The possible values are:

--display=csv

Comma separated data with the unique identifiers as rows.

--display=csvtranspose

Comma separated data with the unique identifiers as columns.

--display=space

Space separated data with the unique identifiers as rows.

--display=transpose

Space separated data with the unique identifiers as columns.

--instrument[=BOOLEAN]

If set then the summary includes the source instrument from the metadata.

--mean[=BOOLEAN]

If set then the summary includes the global arithmetic mean of each output.

--metadata=STRING...

This sets any additional metadata paths to include in the final summary output. Because these are looked up from metadata values they will usually begin with a "^" indicating a metadata path.

--rows=TYPE

This is the method that rows are created using. This determines how data are placed into various output binnings.

Default: **Determined by primary mode**

The possible values are:

--rows=flavorless

Create rows for each station, archive, and variable combination

--rows=instrument

Create a row for each instrument code (e.x. "S11")

--rows=variable

Create rows for each unique station, archive, variable and flavors combination

--stddev[=BOOLEAN]

If set then the summary includes the global standard deviation of each output.

--wavelength[=BOOLEAN]

If set then the summary includes the variable wavelength from the metadata.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Space separated columns by variable

```
da.summary.general bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.summary.general input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.summary.general
```

Comma separated displaying wavelengths, units and formats

```
da.summary.general --cover=false --display=csv --mean=false
--metadata=^Format,^Units --rows=flavorless --wavelength bnd S11a 2015-05-01 2015-
05-03
```

da.summary.timeline

This component provides a means of generating a high level summary timeline of input data. This timeline can provide information about changes to the data and metadata, coverage information, and significant events.

Usage

da.summary.timeline [switches...] [[station] variables times [archive]|[file]]

Switches

--mode=TYPE

This is the base summary mode. This determines the results output as well as the method of displaying them.

Default: **Instrument coverage**

The possible values are:

--mode=changes

Data change events

--mode=cover

Coverage fraction report

--mode=instrument

Instrument coverage summary

--mode=presence

Variable presence summary

--mode=variable

Variable coverage summary

--active=TYPE

When enabled a summary of the currently active parameters is output. For example this would output the instruments active if instrument display is enabled.

Default: **Determined by base mode**

The possible values are:

--active=latest

A summary of active parameters relative to the current time.

--active=none

Latest active display disabled.

--active=normal

A summary of active parameters.

--breakdown=TYPE

This is the method that input data is broken down into separate traces or columns. For example in instrument mode all inputs belonging to the same instrument are combined into one output.

Default: **Determined by base mode**

The possible values are:

--breakdown=flavorless

Break down data into a separate output for each unique combination of station, archive, and variable while combining flavors.

--breakdown=instrument

Combine data for instruments together (e.x. "S11").

--breakdown=variable

Break down data into a separate output for each unique combination of station, archive, variable and flavors.

--cover[=BOOLEAN]

When enabled the timeline will incorporate coverage information into the display.

Default: **Determined by base mode**

--data=**STRING...**

This sets any additional data paths to include in the final summary output.

--data-active[=**BOOLEAN]**

When enabled the current values of the additional data parameters will be incorporated into the output report. This has no effect on symbol based timelines and no effect unless one or more additional data paths are set.

--data-events[=**BOOLEAN]**

When enabled changes to the additional data specifications will be incorporated into the output. Unless one or more additional data paths are set, this will have no effect.

Default: **Enabled**

--display=**INTERVAL**

This is the interval that data are displayed at. For example this sets the interval between lines in CSV report mode or the time in each symbol on a timeline. When not set it defaults to the calculation interval for reports or a fixed amount of time determined by the timeline size.

Default: **Automatic**

The interval must be greater than zero in length.

--events=**TYPE**

This is the way change events are displayed.

Default: **Determined by base mode**

The possible values are:

--events=csv

Comma separated events are displayed.

--events=none

Event output disabled.

--events=normal

Space separated events are displayed.

--events=presence

Space separated presence or absence only events are displayed.

--gap=**INTERVAL**

This is the interval between an event representing absence and the next valid one before anything is displayed. That is, this is them minimum amount of time something must be absent for before the absence is reported.

Default: **One day**

Undefined intervals are accepted. The interval must be greater than or equal to zero length.

--instrument-active[=BOOLEAN]

When enabled active instrument information will be incorporated into the output report. This has no effect on symbol based timelines.

--instrument-events[=BOOLEAN]

When enabled instrument change events will incorporated into the output.

Default: **Determined by base mode**

--interval=INTERVAL

This is the interval that continuous parameters are calculated at. For example, all data are binned together at this interval when calculating coverage. This will also set the display interval if none is explicitly set.

Default: **One day**

The interval must be greater than zero in length.

--legend[=BOOLEAN]

When set a legend is output when displaying a symbol based timeline.

Default: **Enabled**

--metadata=STRING...

This sets any additional data paths to include in the final summary output. Because these are looked up from metadata values they will usually begin with a "^" indicating a metadata path

--metadata-active[=BOOLEAN]

When enabled the current values of the additional data parameters will be incorporated into the output report. This has no effect on symbol based timelines and no effect unless one or more additional data paths are set.

--metadata-events[=BOOLEAN]

When enabled changes to the additional data specifications will be incorporated into the output. Unless one or more additional data paths are set, this will have no effect.

Default: **Enabled**

--presence-active[=BOOLEAN]

When enabled current presence information will be incorporated into the output report. This has no effect on symbol based timelines.

--presence-events[=BOOLEAN]

When enabled presence change events will incorporated into the output.

Default: **Determined by base mode**

--show-interval[=BOOLEAN]

When enabled data interval information will be incorporated into the output report. This has no effect on symbol based timelines.

--timeline=TYPE

This is the way the timeline is displayed.

Default: **Determined by base mode**

The possible values are:

--timeline=csv

A CSV style time series report is generated with the values at each time step reported as rows and with the outputs as columns.

--timeline=horizontal

A timeline of symbols for each output is generated with the time axis as columns and the outputs as rows.

--timeline=none

Timeline output disabled.

--timeline=vertical

A timeline of symbols for each output is generated with the time axis as rows and the outputs as columns.

--wavelength-active[=BOOLEAN]

When enabled current wavelength information will be incorporated into the output report. This has no effect on symbol based timelines.

--wavelength-events[=BOOLEAN]

When enabled wavelength change events will incorporated into the output.

Default: **Determined by base mode**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Instrument and coverage information

```
da.summary.timeline bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.summary.timeline input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.summary.timeline
```

Vertical instrument information at quarter resolution

```
da.summary.timeline --interval=1qa --timeline=vertical bnd S11a 2015-05-01 2015-05-
03
```

Coverage and data interval report

```
da.summary.timeline --mode=cover --show-interval bnd S11a 2015-05-01 2015-05-03
```

Wavelength change events

```
da.summary.timeline --mode=changes --instrument-events=false --wavelength-events
bnd S11a 2015-05-01 2015-05-03
```

da.where

This component provides means of selecting data for output using small segments of script code. This primarily takes the form of specifying a condition where data is included in or excluded from the output.

Usage

da.where [switches...] [[station] variables times [archive]|[file]]

Switches

--fanout-archive[=BOOLEAN]

If set then the condition is instantiated uniquely for each archive (excluding metadata).

Default: **Enabled**

--fanout-flavors[=BOOLEAN]

If set then the condition is instantiated uniquely for each combination of flavors (e.x. PM1 vs PM10). This also automatically excludes information flavors like statistics.

Default: **Enabled**

--fanout-station[=BOOLEAN]

If set then the condition is instantiated uniquely for each station (exclude the default, if any).

Default: **Enabled**

--fanout-variable[=BOOLEAN]

If set then the condition is instantiated uniquely for variable.

--include=SCRIPT

This sets the code executed at each data step. The test is evaluated as segments called 'data', so to

test a variable the syntax results in code like: 'return data.BsG_S11 > 2.0'

--inputs=SELECTION

This option sets the inputs used in the script condition.

Default: **Everything**

--invert[=BOOLEAN]

Invert the condition and apply the effect when it is not met.

--outputs=SELECTION

This option sets the data that is removed or undefined when the condition is met.

Default: **Everything**

--undefine[=BOOLEAN]

When set values that fail the conditions are set to undefined instead of being removed from the data stream entirely.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Simple condition

This selects all data for which BsG_S11 is greater than 2.0 and is greater than BsR_S11. If either are not defined then no output is produced.

```
da.where --include='return data.BsG_S11 > 2.0 and data.BsG_S11 > data.BsR_S11' bnd
S11a 2015-05-01 2015-05-03
```

Or:

```
da.where --include='return data.BsG_S11 > 2.0 and data.BsG_S11 > data.BsR_S11'
input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.where --include='return data.BsG_S11 >
2.0 and data.BsG_S11 > data.BsR_S11'
```

Zero multiple

This selects all data for which either BsG_S11 or BsR_S11 is exactly zero. If one is undefined but the other is zero then output is still produced.

```
da.where --include='return data.BsG_S11 == 0 or data.BsR_S11 == 0' bnd S11a 2015-
05-01 2015-05-03
```

Time condition

This excludes all data between UTC hour 8 and 12.

```
da.where --include='hour = (data.START % 86400) / 3600; return hour >= 8 and hour <
12' --invert bnd S11a 2015-05-01 2015-05-03
```

da.multiplex

This program is used to combine multiple input data sources into a single output stream. The most common use is combining multiple commands into a single output for a pipeline. Output is written to standard output for pipeline usage.

Usage

da.multiplex <input ...>

Arguments

Each argument specifies an input to be combined into the final output. Unless a special condition is met, the input is treated as a file name to read data from.

The accepted special conditions are:

The argument starts with `|` (a pipe)

It is treated as a command evaluated with the shell. This mode allows multiple commands to be multiplexed together to create a final output.

The argument is exactly `-` (a single dash)

The multiplexer will also read standard input and combine that with any other explicit inputs. This mode allows it to be used to multiplex streams as part of a pipeline.

The argument is exactly `--` (two dashes)

The multiplexer will ignore the `--` but treat all further arguments as files, regardless of the special conditions above.

Examples

Combining files

This multiplexes two files together to generate an output.

```
da.mulitplex first_file.c3d second_file.c3d > combined.c3d
```

Combining commands

This combines two commands into a single output.

```
da.multiplex '| da.get bnd BsG_S11 2016-05-01 1d' '| da.get mlo BsG_S11 2015-05-01
1d' | da.export
```

Standard input and a file

This combines standard input an a file into the output

```
da.get bnd BsG_S11 2016-05-01 1d | da.multiplex - -- first_file.c3d
```

Correction and Calculation

These programs process data to apply corrections or calculate new derived parameters. They are generally used as part of a data pipeline.

da.calc.dewpoint

This component calculated dewpoint, temperature, or RH from the two that are present. That is, if given a temperature and RH it can calculate the dewpoint for those measurements.

Usage

da.calc.dewpoint [switches...] [[station] variables times [archive]|[file]]

Switches

--always[=**BOOLEAN]**

When set, this enables calculation of the outputs regardless of if they already have values. Normally the outputs are only calculated when they do not already exist in the data, but this will cause them to always be (re-)calculated.

--input-rh=**INPUT**

This is the relative humidity to use in calculations. This is used when calculating a dewpoint or a temperature.

Default: **Resolved automatically from other variables**

--input-t=**INPUT**

This is the temperature to use in calculations. This is used when calculating a relative humidity or dewpoint.

Default: **Resolved automatically from other variables**

--input-td=**INPUT**

This is the dewpoint to use in calculations. This is used when calculating a relative humidity or temperature.

Default: **Resolved automatically from other variables**

--instruments-rh=**SUFFIX...**

These are the instrument suffixes to perform relative humidity calculation for. When set, only relative humidities with these suffixes are generated. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--instruments-t=**SUFFIX...**

These are the instrument suffixes to perform temperature calculation for. When set, only

temperatures with these suffixes are generated. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--instruments-td=SUFFIX...

These are the instrument suffixes to perform dewpoint calculation for. When set, only dewpoints with these suffixes are generated. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--output-rh=SELECTION

These are the relative humidity variables that are generated. When only these variables are calculated from a single set of inputs.

Default: **All relative humidities**

--output-t=SELECTION

These are the temperature variables that are generated. When only these variables are calculated from a single set of inputs.

Default: **All temperatures**

--output-td=SELECTION

These are the dewpoint variables that are generated. When only these variables are calculated from a single set of inputs.

Default: **All temperatures**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That

is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will calculate temperatures, dewpoints, and relative humidities for all sensors present in the input data stream. If two are present the missing third one will be calculated. For example, if the input contains T_V11 and U_V11 the output will have TD_V11 generated.

```
da.calc.dewpoint bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.calc.dewpoint input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.calc.dewpoint
```

Single instrument with alternate humidity

This will calculate the dewpoints for all sensors belonging to the S11 instrument but using the value of U_V11 as the relative humidity.

```
da.calc.dewpoint --input-rh=bnd:raw:U_V11 --instruments-td=S11 bnd S11a 2015-05-01
2015-05-03
```

Single variable

This will calculate U_XM1 from TD_XM1 and T_XM1.

```
da.calc.dewpoint --input-t=bnd:raw:T_XM1 --input-td=bnd:raw:TD_XM1 --output
-rh=bnd:raw:U_XM1 bnd S11a 2015-05-01 2015-05-03
```

da.calc.integratesize

This calculates integrated properties from a size distribution. For example, it calculates Mie scattering as well as number and volume concentrations.

Usage

da.calc.integratesize [switches...] [[station] variables times [archive]|[file]]

Switches

--angles=INTEGER

The number of angles between 0 and 90 degrees used in Mie calculations.

Default: **2**

This option only accepts integers greater than or equal to 2.

--ba=NUMBER...

These are the wavelengths to calculate absorption at. When set to empty, no absorptions are generated.

Default: **Disabled**

This option only accepts numbers strictly greater than 0.

--bbs=NUMBER...

These are the wavelengths to calculate back scattering at. When set to empty, no back scatterings are generated.

Default: **450,550,700**

This option only accepts numbers strictly greater than 0.

--be=NUMBER...

These are the wavelengths to calculate extinction at. When set to empty, no back extinctions are generated.

Default: **Disabled**

This option only accepts numbers strictly greater than 0.

--bs=NUMBER...

These are the wavelengths to calculate scattering at. When set to empty, no scatterings are generated.

Default: **450,550,700**

This option only accepts numbers strictly greater than 0.

--bsfm=NUMBER...

These are the wavelengths to calculate the fine mode scattering at. The diameter cut off used to determine what aerosol is fine mode is set with --fine-diameter. When set to empty, no fine mode scatterings are generated.

Default: **Disabled**

This option only accepts numbers strictly greater than 0.

--dnb[=BOOLEAN**]**

When enabled the un-normalized distribution (dN) is calculated.

Default: **Disabled**

--dnn[=BOOLEAN**]**

When enabled the normalized distribution (dN/dlogDp) is calculated.

Default: **Disabled**

--ds=NUMBER...

These are the wavelengths to calculate generate the scattering distribution (dσ/dLogDp) at.

Default: **550**

This option only accepts numbers strictly greater than 0.

--dv[=BOOLEAN**]**

When enabled the volume distribution (dV/dlogDp) is calculated.

Default: **Enabled**

--fine-diameter=NUMBER

The maximum diameter bin in µm to include in calculations of fine mode only properties. All bins greater than this diameter are excluded from the integration.

Default: **0.7 µm**

This option only accepts numbers strictly greater than 0.

--g=NUMBER...

These are the wavelengths to calculate the asymmetry parameter at. When set to empty, no asymmetry parameters are generated.

Default: **Disabled**

This option only accepts numbers strictly greater than 0.

--instruments=SUFFIX...

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--maximum-diameter=NUMBER

The maximum diameter bin in µm to include in any calculations. All bins greater than this are excluded from all calculations.

This option only accepts numbers strictly greater than 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--minimum-diameter=NUMBER

The minimum diameter bin in µm to include in any calculations. All bins less than this are excluded from all calculations.

This option only accepts numbers strictly greater than 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--n[=BOOLEAN]

When enabled the total number concentration is calculated.

Default: **Enabled**

--nm[=BOOLEAN]

When enabled the geometric number mean diameter and standard deviation are calculated.

Default: **Enabled**

--nmv[=BOOLEAN]

When enabled the volume mean diameter is calculated.

Default: **Enabled**

--nv[=BOOLEAN]

When enabled the total volume concentration is calculated.

Default: **Enabled**

--ri-i=NUMBER

The imaginary component of the refractive index used in Mie calculations.

Default: **0.001**

This option only accepts numbers strictly greater than 0.

--ri-r=NUMBER

The real component of the refractive index used in Mie calculations.

Default: **1.53**

This option only accepts numbers strictly greater than 0.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will generate all default size distribution integrations from any size distributions in the input data stream.

```
da.calc.integratesize bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.calc.integratesize input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.calc.integratesize
```

Single instrument with manual input list

This will calculate the size distributions from N11 only.

```
da.calc.integratesize --instruments=N11 bnd S11a 2015-05-01 2015-05-03
```

da.calc.intensives

This calculates intensive optical intensive properties from its input. This include name normalization, single scattering albedos, Ångström exponents, asymmetry parameters, and radiative forcing efficiency.

Usage

da.calc.intensives [switches...] [[station] variables times [archive]|[file]]

Switches

--absorption=**SELECTION**

This is the input light absorption to use in intensives generation. This option is mutually exclusive with absorption instrument specification.

Default: **All light absorptions**

--absorption-instruments=**SUFFIX...**

These are the instrument suffixes to use absorption data from. For example A11 would usually specifies the primary absorption instrument. This option is mutually exclusive with manual absorption variable specification.

Default: **All instrument suffixes**

--angstrom=**INPUT**

This is the Ångström exponent used to perform wavelength adjustment when interpolation is not possible. When not set, data are simply left as undefined when they cannot be adjusted to the target wavelengths.

--angstrom-distance=**NUMBERS**

This is the maximum distance that the fallback Ågström exponent is valid for. This is used in conjunction with angstrom-wavelength to limit how far data are extrapolated.

This option only accepts numbers greater than or equal to 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--angstrom-wavelength=**NUMBERS**

This is the wavelength of the Ångström exponent used to perform wavelength adjustment when interpolation is not possible. This is used to limit how far data are extrapolated when used in conjunction with angstrom-distance.

This option only accepts numbers strictly greater than 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--backscattering=**SELECTION**

This is the input light backscattering to use in intensives generation. This option is mutually exclusive with scattering instrument specification.

Default: **All light backscatterings**

--bfr=**INPUT**

When set, this specifies a backscatter fraction to use. This allows other parameters to be calculated when there is no backscattering data available by assuming a fixed backscatter ratio.

Default: **Calculated from the scattering data**

--counts=**INPUT**

This is the input concentration to forward and rename in the output. This option is mutually

exclusive with concentration instrument specification.

Default: **All concentrations**

--counts-instruments=**SUFFIX...**

These are the instrument suffixes to use particle concentrations from. For example N61 or N71 usually specifies the system CPC. This option is mutually exclusive with manual counts variable specification.

Default: **All instrument suffixes**

--extinction=**SELECTION**

This is the input light extinction to use in intensives generation. This option is mutually exclusive with extinction instrument specification.

Default: **All light extinctions**

--extinction-instruments=**SUFFIX...**

These are the instrument suffixes to use extinction data from. For example E11 would usually specifies the first extinction instrument. This option is mutually exclusive with manual extinction variable specification.

Default: **All instrument suffixes**

--mac=**INPUT**

This is the calibration factor used to calculate equivalent black carbon concentrations. The efficiency for each wavelength is this value divided by the wavelength in nm.

Default: **14625**

--output=**SUFFIX...**

These are the instrument suffixes that output data will be generated with first. If the number of outputs exceeds the number of suffixes defined then the defaults are used.

Default: **XI, XJ, etc**

--scattering=**SELECTION**

This is the input light scattering to use in intensives generation. This option is mutually exclusive with scattering instrument specification.

Default: **All light scatterings**

--scattering-instruments=**SUFFIX...**

These are the instrument suffixes to use scattering data from. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual scattering variable specification.

Default: **All instrument suffixes**

--ssa=**INPUT**

When set, this specifies a single scattering albedo to use. This allows other parameters to be calculated assuming a fixed value.

Default: **Calculated from data**

--wavelengths=NUMBER...

These are the output wavelengths to calculate the intensive parameters at. Input data will be adjusted to these wavelengths before use.

Default: **450,550,700**

This option only accepts numbers strictly greater than 0.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an

archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will generate intensives for all input data. Note that the output variable assignment is unordered in the case of multiple sources of extensive parameters,

```
da.calc.intensives bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.calc.intensives input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.calc.intensives
```

Restricted instruments

This will calculate intensives from only S12, A12, and N61 regardless of what data is present in the input. Output variables will have the XJ suffix (for example BsG_XJ).

```
da.calc.intensives --absorption-instruments=A12 --counts-instruments=N61
--extinction-instruments= --output=XJ --scattering-instruments=S12 bnd S11a 2015-
05-01 2015-05-03
```

Non-standard wavelengths and manual backscatter fraction

This will calculate intensives at 467, 528, and 652nm using a fixed intensives fraction of 0.2 instead of calculating it from the data.

```
da.calc.intensives --bfr=0.2 --wavelengths=467,528,652 bnd S11a 2015-05-01 2015-05-
03
```

Explicit scattering

This will calculate intensives using only green scattering from the S11 instrument.

```
da.calc.intensives --backscattering=bnd:raw:BbsG_S11 --scattering=bnd:raw:BsG_S11
bnd S11a 2015-05-01 2015-05-03
```

da.calc.lwcprobe

This component calculates the liquid water concentration based on a heating element probe. It takes the heating power, air and sensor temperatures, air speed, and pressure to produces a concentration of liquid water.

Usage

da.calc.lwcprobe [switches...] [[station] variables times [archive]|[file]]

Switches

--input-air-p=**INPUT**

This is the air pressure to use in calculations.

Default: **1013.25 hPa**

--input-air-speed=**INPUT**

This is the air speed past the sensor used in calculations.

Default: **1 m/s**

--input-air-t=**INPUT**

This is the air temperature to use in calculations. This is the ambient air temperature being sampled.

Default: **273.15 K**

--input-power=**INPUT**

This is the power in watts being consumed by the sensing element. This option is mutually exclusive with with instrument specification.

Default: **All powers**

--input-probe-diameter=**INPUT**

This is the diameter of the sensing element in the probe.

Default: **0.18 cm**

--input-probe-length=**INPUT**

This is the length of the sensing element in the probe.

Default: **2.0 cm**

--input-probe-temperature=INPUT

This is temperature the probe sensing element is being maintained at.

Default: **125 °C**

--instruments=SUFFIX...

These are the instrument suffixes to calculate concentrations for. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--output-concentration=SELECTION

This is the output concentration that is calculated. This option is mutually exclusive with with instrument specification.

Default: **Calculated from input powers**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data

are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will calculate concentrations for all power readings in the input using assumed defaults. For example, it will calculate X_N11 from VA_N11.

```
da.calc.lwcprobe bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.calc.lwcprobe input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.calc.lwcprobe
```

Single instrument with measured temperature and pressure

This will calculate the concentration for the N11 instrument using temperature T_V01 and the pressure P_XM and assumed air speed.

```
da.calc.lwcprobe --input-air-p=bnd:raw:P_XM --input-air-speed=10 --input-air
-t=bnd:raw:T_V01 --instruments=N11 bnd S11a 2015-05-01 2015-05-03
```

Single variable

This will calculate X_X1 from VA_XM for assumed probe parameters.

```
da.calc.lwcprobe --input-power=bnd:raw:VA_XM --output-concentration=bnd:raw:X_X1
bnd S11a 2015-05-01 2015-05-03
```

da.calc.opticaldepth

This calculates optical depths by integrating extensive parameters over the course of a filter. It can calculate scattering, absorption and extinction optical depths for the filter in addition to the asymmetry parameter derived from the backscattering optical depth. This is generally used as a precursor to applying the Müller 2014 (CTS) correction.

Usage

da.calc.opticaldepth [switches...] [[station] variables times [archive]|[file]]

Switches

--absorption=SELECTION

This is the input light absorption to use in optical depth generation. This option is mutually exclusive with absorption instrument specification.

Default: **All light absorptions**

--absorption-instruments=SUFFIX...

These are the instrument suffixes to use absorption data from. For example A11 would usually specifies the primary absorption instrument. This option is mutually exclusive with manual absorption variable specification.

Default: **All instrument suffixes**

--backscattering=SELECTION

This is the input light backscattering to use in optical depth generation. This option is mutually exclusive with scattering instrument specification.

Default: **All light backscatterings**

--calculate-l[=BOOLEAN**]**

This enables recalculation of the sample length from the spot area and flow. If the variables containing the spot area and flow are not present in the input then the option to set the incremental change in length most be set.

--dl=INPUT

This is the change in sample length used for each calculation interval. When omitted it is calculated from the start and end sample lengths.

--extinction=SELECTION

This is the input light extinction to use in optical depth generation. This option is mutually exclusive with extinction instrument specification.

Default: **All light extinctions**

--extinction-instruments=SUFFIX...

These are the instrument suffixes to use extinction data from. For example E11 would usually

specifies the first extinction instrument. This option is mutually exclusive with manual extinction variable specification.

Default: **All instrument suffixes**

--instruments=**SUFFIX...**

These are the instrument suffixes to calculate for. For example S11 would usually specifies the reference nephelometer.

Default: **All instrument suffixes**

--length-end=**INPUT**

This is the input end sample length used in the calculation. When this option is set, this length is used for all channels.

Default: **Resolved automatically by instrument suffix**

--length-start=**INPUT**

This is the input start sample length used in the calculation. When this option is set, this length is used for all channels.

Default: **Resolved automatically by instrument suffix**

--reset-gap=**INTERVAL**

This is the maximum time allowed between accumulated measurements. When this time is exceeded, the integration is reset (a new filter begins). Invalid but present measurements are not considered gaps.

Undefined intervals are accepted. The interval must be greater than or equal to zero length.

--scattering=**SELECTION**

This is the input light scattering to use in optical depth generation. This option is mutually exclusive with scattering instrument specification.

Default: **All light scatterings**

--scattering-instruments=**SUFFIX...**

These are the instrument suffixes to use scattering data from. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual scattering variable specification.

Default: **All instrument suffixes**

--synchronous-zero[=**BOOLEANS]**

When enabled, this option causes zero change to the integrated optical depths when their corresponding instrument reports a zero in its flags. When not present the last known value continues to be integrated. This is normally used when the filter based instrument is sampling from a pickoff downstream of the zeroing instrument.

--transmittance=SELECTION

This is the input transmittance used to detect filter changes. When set this is used for all channels and all instruments so it normally only makes sense for a single channel.

Default: **Resolved automatically by instrument suffix**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will calculate all optical depths for any instrument that provides a transmittance using any available extensive parameters.

```
da.calc.opticaldepth bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.calc.opticaldepth input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.calc.opticaldepth
```

Restricted instruments

This will calculate scattering optical depth from S12 and absorption optical depth from A12 based on the filter of A11 regardless of what data is present in the input.

```
da.calc.opticaldepth --absorption-instruments=A12 --extinction-instruments=
--instruments=A11 --scattering-instruments=S12 bnd S11a 2015-05-01 2015-05-03
```

Explicit scattering

This will calculate optical depth using only green scattering from the S11 instrument.

```
da.calc.opticaldepth --backscattering=bnd:raw:BbsG_S11 --scattering=bnd:raw:BsG_S11
bnd S11a 2015-05-01 2015-05-03
```

da.calc.pitotflow

This component calculates flow from a pressure difference measured with a pitot tube. That is, given a temperature, absolute pressure, cross-sectional area, and delta pressure it calculates the flow rate.

Usage

da.calc.pitotflow [switches...] [[station] variables times [archive]|[file]]

Switches

--diameter=NUMBER

This is the diameter of the flow tube in mm. This is used to set the cross sectional duct area for calculating the flow.

Default: **44.45 mm = 1.75 in**

This option only accepts numbers strictly greater than 0.

--input-area=INPUT

This is cross sectional area of the duct at the pitot tube in m². This option is mutually exclusive with the diameter.

--input-dp=INPUT

This is the delta pressure at the pitot tube. This option is mutually exclusive with with instrument specification.

Default: **All delta pressures**

--input-p=INPUT

This is the temperature to use in calculations. This is the pressure of the air at the pitot tube.

Default: **1013.25 hPa**

--input-t=INPUT

This is the temperature to use in calculations. This is the temperature of the air at the pitot tube.

Default: **273.15 K**

--instruments=SUFFIX...

These are the instrument suffixes to calculate flows for. For example P01 would usually specifies the stack flow pitot sensor. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--output-q=SELECTION

This is the output flow that is calculated. This option is mutually exclusive with with instrument specification.

Default: **Calculated from input pressures**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular

expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will calculate flows for all delta pressures in the input assuming they are at STP with 1.75 in ID tubing. For example, it will calculate Q_P01 from Pd_P01.

```
da.calc.pitotflow bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.calc.pitotflow input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.calc.pitotflow
```

Single instrument with measured temperature and pressure

This will calculate the flow for the P01 sensor using the measured temperature T_V01 and the pressure P_XM assuming a duct area of 0.002 m².

```
da.calc.pitotflow --input-area=0.002 --input-p=bnd:raw:P_XM --input-t=bnd:raw:T_V01
--instruments=P01 bnd S11a 2015-05-01 2015-05-03
```

Single variable

This will calculate Q_X1 from Pd_XM for a flow tube with a 30 mm inner diameter.

```
da.calc.pitotflow --diameter=30 --input-dp=bnd:raw:Pd_XM --output-q=bnd:raw:Q_X1
bnd S11a 2015-05-01 2015-05-03
```

da.calc.purpleair

This component calculates the derived parameters for a PurpleAir instrument. It takes the raw concentration intensities and produces a combined output intensity as well as a scattering from the combined intensity.

Usage

da.calc.purpleair [switches...] [[station] variables times [archive]|[file]]

Switches

--efficiency=COEFFICIENTS...

This is the calibration applied to the combined intensity to generate the output light scattering value.

Default: **Intensity/55**

--input-a=INPUT

This is the count reading from detector A.

Default: **All available detectors**

--input-b=INPUT

This is the count reading from detector B.

Default: **All available detectors**

--instruments=**SUFFIX...**

These are the instrument suffixes to calculate combined outputs for. This option is mutually exclusive with manual variable specification.

Default: **All scattering instrument suffixes**

--output-intensity=**SELECTION**

This is the output for the combined intensity created from the weighted average of the input detector readings.

Default: **Calculated from input intensities**

--output-scattering=**SELECTION**

This is the output for the combined scattering created by applying the scattering efficiency to the combined detector reading.

Default: **Calculated from input intensities**

--wavelength=**NUMBERS**

This is the wavelength set in the metadata for the output light scattering.

Default: **550 nm**

--weight-a=**INPUT**

This is the relative weight applied to the intensity reading from detector A. The combined intensity is generated by applying the relative weights to the input intensities and normalizing by the total weight.

Default: **1.0 unless set in metadata**

--weight-b=**INPUT**

This is the relative weight applied to the intensity reading from detector B. The combined intensity is generated by applying the relative weights to the input intensities and normalizing by the total weight.

Default: **1.0 unless set in metadata**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will calculate combined intensities and scatterings for all PurpleAir intensity inputs. For example, it will calculate Ip_S81 and Bs_S81 from Ipa_S81 and Ipb_S81. The weights used to combine the intensities are read from the metadata or assumed to be 1.0 if not included in the metadata.

```
da.calc.purpleair bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.calc.purpleair input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.calc.purpleair
```

Single instrument ignoring detector A

This will calculate the outputs, using only detector B for the measured intensities.

```
da.calc.purpleair --instruments=S91 --weight-a=0 bnd S11a 2015-05-01 2015-05-03
```

Single variable

This will calculate BsG_S81 and Ip_S81 from Ipa_S82 and Ipa_S81 using the default weights.

```
da.calc.purpleair --input-a=bnd:raw:Ipa_S82 --output-scattering=bnd:raw:BsG_S81 bnd
S11a 2015-05-01 2015-05-03
```

da.calc.radiation

This component performs the calibration and calculation for surface radiation data. It takes the raw units (e.g. mV) and produces scientific units (e.g. W/m²) using calibration factors.

Usage

da.calc.radiation [switches...] [[station] variables times [archive]|[file]]

Switches

--type=TYPE

This is the calibration performed on the inputs to generate the radiation intensity.

Default: **Division**

The possible values are:

--type=BW

Same as --type=**Division**

--type=CGR4

Same as --type=**Pyrgeometer**

--type=CHP1

Same as --type=**Division**

--type=CM22

Same as --type=**Division**

--type=Division

Division by the calibration factor

--type=EppleyBW

Same as --type=**Division**

--type=EppleyNIP

Same as --type=**Division**

--type=EppleyPIR

Same as --type=**Pyrgeometer**

--type=EppleyPSP

Same as --type=**Division**

--type=KippZonenCGR4

Same as --type=**Pyrgeometer**

--type=KippZonenCHP1

Same as --type=**Division**

--type=KippZonenCM22

Same as --type=**Division**

--type=Multiplication

Multiplication by the calibration factor

--type=NIP

Same as --type=**Division**

--type=PIR

Same as --type=**Pyrgeometer**

--type=PSP

Same as --type=**Division**

--type=Pyrgeometer

Pyrgeometer calculation (R=V*cal+σ*T)

--case=INPUT

This is case temperature used for pyrgeometer corrections.

Default: **Automatically selected**

--dcf=INPUT

This is the dome temperature correction factor. This is applied to the difference of the case and dome temperatures each to the fourth, before the Stefan–Boltzmann constant.

Default: **Disabled**

--dome=**INPUT**

This is dome temperature used for pyrgeometer corrections.

Default: **Automatically selected**

--e=**INPUT**

This is the E factor used for case temperature correction. This is applied to the case temperature before the Stefan–Boltzmann constant.

Default: **1**

--factor=**INPUT**

This is the calibration factor applied to the input. This factor should convert the input sensor reading to W/m².

--input=**INPUT**

This is the input sensor reading. This option is mutually exclusive with with instrument specification.

Default: **All radiation voltages**

--instruments=**SUFFIX...**

These are the instrument suffixes to calculate temperatures for. This option is mutually exclusive with manual variable specification.

Default: **All radiation instruments**

--output=**SELECTION**

This is output radiation intensity. This option is mutually exclusive with with instrument specification.

--sigma=**INPUT**

This is the Stefan–Boltzmann constant used.

Default: **5.670374419E-8 W/m²K**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of

selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

da.calc.resistancetemperature

This component performs calculation of temperature from a resistance measurement.

Usage

da.calc.resistancetemperature [switches...] [[station] variables times [archive]|[file]]

Switches

--type=TYPE

This is the calculation performed on the inputs to generate the temperature.

Default: **Steinhart-Hart**

The possible values are:

--type=Disable calculation

No calculation applied

--type=PT100

Same as --type=**rtd**

--type=PT1000

Same as --type=**rtd**

--type=Steinhart-Hart

Steinhart-Hart with A, B, and C coefficients

--type=ntc

Simple NTC thermistor with R =A, β=B, T =C (default 25)

--type=rtd

Linear RTD with R =A, α=B, T =C (default 0)

--a=INPUT

The first input. The usage depends on the calculation selected.

Default: **Calculation dependent**

--b=INPUT

The second input. The usage depends on the calculation selected.

Default: **Calculation dependent**

--c=INPUT

The third input. The usage depends on the calculation selected.

Default: **Calculation dependent**

--calibration=COEFFICIENTS...

The calibration applied to the input and any same unit types used in the calculation.

Default: **Conversion to ohms when required**

--instruments=SUFFIX...

These are the instrument suffixes to calculate temperatures for. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--output=SELECTION

This is output temperature. This option is mutually exclusive with with instrument specification.

--resistance=INPUT

This is the input resistance. This option is mutually exclusive with with instrument specification.

Default: **All resistances**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

da.calc.simple

This component performs simple calculations on its inputs to produce a calculated output value. Example calculations include summation, multiplication, division and other simple operations.

Usage

da.calc.simple [switches...] [[station] variables times [archive]|[file]]

Switches

--operation=TYPE

This is the operation performed on the inputs to generate the output.

Default: **None**

The possible values are:

--operation=Add

A + B

--operation=AddAll

A + B + C

--operation=AddMulitply

A * (B + C)

--operation=CosScaled

cos(A) * B

--operation=Divide

A / B

--operation=DivideAll

A / B / C

--operation=Identity

Same as --operation=**None**

--operation=Limit

min(max(A, B), C) (i.e. A if B ≤ A ≤ C)

--operation=Log

Same as --operation=**Logarithm**

--operation=Logarithm

log_B(A) (i.e. log(A)/log(B))

--operation=Max

max(A, B) (i.e. A if A > B)

--operation=Min

min(A, B) (i.e. A if A < B)

--operation=Multiply

A * B

--operation=MultiplyAdd

A * B + C

--operation=MultiplyAll

A * B * C

--operation=None

A

--operation=Power

A

--operation=RatioSum

A / (B + C)

--operation=SinScaled

sin(A) * B

--operation=Subtract

A - B

--operation=SubtractAll

A - B - C

--operation=SumRatio

(A + B) / C

--operation=UndefinedAbove

A if A ≤ B

--operation=UndefinedBelow

A if A ≥ B

--operation=UndefinedOutside

A if B ≤ A ≤ C

--operation=Wrap

Wrap A in [0,B]

--operation=WrapRange

Wrap A in [B,C]

--operation=abs

|A|

--operation=acos

acos(A)

--operation=asin

asin(A)

--operation=atan

atan(A)

--operation=atan2

atan2(A,B)

--operation=cos

cos(A)

--operation=exp

e

--operation=ln

ln(A)

--operation=log10

log (A)

--operation=sin

sin(A)

--operation=tan

tan(A)

--a=INPUT

The first input. The usage depends on the operation selected.

--b=INPUT

The second input. The usage depends on the operation selected.

--c=INPUT

The third input. The usage depends on the operation selected.

--calibration=**COEFFICIENTS...**

The calibration applied to the output after the calculation is performed.

--fanout-variable=**STRING**

When set, this matches input variables against the given regular expression. When one matches, a new copy of the operation is created with the other parameters (station, archive, flavors). This is most commonly used to handle cut-size selected data by matching one of the inputs that is size selected.

Default: **Disabled**

--inverse-calibration[=**BOOLEANS]**

When set, the calibration is inverted (backed out) instead of applied. This is used to get the original values of an already applied calibration.

--output=**SELECTION**

This is output value of the calculation.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Calculate non-zero adjusted scattering

This calculates BsG_S11 + BswG_S11 and places the output into ZBsfG_S11. That is, it generates the total scattering with the wall scattering added back in.

```
da.calc.simple --operation=Add --a=bnd:raw:BsG_S11 --b=bnd:raw:BswG_S11
--output=bnd:raw:ZBsfG_S11 bnd S11a,BswG_S11 2015-05-01 2015-05-03
```

Or:

```
da.calc.simple --operation=Add --a=bnd:raw:BsG_S11 --b=bnd:raw:BswG_S11
--output=bnd:raw:ZBsfG_S11 input_file.c3d
```

Or:

```
da.get bnd S11a,BswG_S11 2015-05-01 2015-05-03 | da.calc.simple --operation=Add
--a=bnd:raw:BsG_S11 --b=bnd:raw:BswG_S11 --output=bnd:raw:ZBsfG_S11
```

da.calc.solarposition

This component calculates solar position parameters for the input data. This includes dark and solar noon flagging in addition to solar position (azimuth and elevation) calculations.

Usage

da.calc.solarposition [switches...] [[station] variables times [archive]|[file]]

Switches

--dark=INPUT

This is the number of degrees below the horizon the sun must be for it to be considered dark and the flag set.

Default: **0**

--flags=SELECTION

This is flags variable that the solar state flags are added to. The calculation adds flags for dark (below the horizon) and a flag on the data point nearest solar noon.

Default: **All available flags variables**

--instruments=SUFFIX...

These are the instrument suffixes to calculate flows for. For example P01 would usually specifies the stack flow pitot sensor. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--latitude=INPUT

This is the latitude in degrees north to calculate parameter for.

Default: **Automatic from data**

--longitude=INPUT

This is the longitude in degrees east to calculate parameter for.

Default: **Automatic from data**

--magnitude=INPUT

This is the magnitude set for the solar position vector. This value is used when data are averaged.

Default: **1**

--output-azimuth=SELECTION

This is the output that contains the calculated solar azimuth angle.

Default: **Automatic if enabled**

--output-elevation=SELECTION

This is the output that contains the calculated solar elevation angle.

Default: **Automatic if enabled**

--output-magnitude=SELECTION

This is the output that contains the specified vector magnitude for use in averaging.

Default: **Automatic if enabled**

--position[=BOOLEAN]

This enables the generation of variables that contain the solar position. This is the current azimuth and elevation angles as well as a magnitude vector used in averaging.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single

dash) may be used to explicitly specify standard input.

Examples

Defaults

This will generate solar noon and dark flagging for all available input flags.

```
da.calc.solarposition bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.calc.solarposition input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.calc.solarposition
```

Single instrument with position

This will generate flagging for only the instrument P01 and add solar position variables to the output.

```
da.calc.solarposition --instruments=P01 --position bnd S11a 2015-05-01 2015-05-03
```

Single position generation

This will the solar position for T_X1 and generate the vector components using it, assuming explicitly specified location coordinates.

```
da.calc.solarposition --latitude=40 --longitude=-105 --magnitude=bnd:raw:T_X1
--output-azimuth=bnd:raw:ZAz_X1 --output-elevation=bnd:raw:ZEl_X1 --output
-magnitude=bnd:raw:ZMa_X1 bnd S11a 2015-05-01 2015-05-03
```

da.corr.andersonogren1998

This component provides a mechanism to apply the Anderson and Ogren 1998 TSI 3563 nephelometer truncation correction.

Usage

da.corr.andersonogren1998 [switches...] [[station] variables times [archive]|[file]]

Switches

--mode=TYPE

This is the correction mode. This determined which set or sets of coefficients to apply. For example, this option can be used to correct data as if it where PM1 (to handle very fine, but not size selected data) or to use the maximum correction factor available (to handle multimodal data).

Default: **Automatic**

The possible values are:

--mode=PM1

Always use PM1 fine mode constants

--mode=PM10

Always use PM10 coarse mode constants

--mode=automatic

Automatically select the constants to use

--mode=automatic

Same as --mode=**automatic**

--mode=coarse

Same as --mode=**PM10**

--mode=fine

Same as --mode=**PM1**

--mode=max

Same as --mode=**maximum**

--mode=maximum

Use the constants that result in the largest correction ratio

--mode=min

Same as --mode=**minimum**

--mode=minimum

Use the constants that result in the smallest correction ratio

--mode=noang

Ignore Ångström exponent fit

--mode=total

Same as --mode=**PM10**

--backscatterings=SELECTION

These are the backscattering variables to correct. This option is mutually exclusive with

instrument specification.

Default: **All backscatterings**

--instruments=SUFFIX...

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--scatterings=SELECTION

These are the scattering variables to correct. This option is mutually exclusive with instrument specification.

Default: **All scatterings**

--smoothing=SMOOTHER

This is the smoothing applied to the scatterings before calculation of the Ångström exponent.

Default: **3-minute TC single pole low pass**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will correct all scatterings and backscatterings using the default smoothing. The constants used in the fits and no-angstrom multiplies are those described in Anderson and Ogren (1998).

```
da.corr.andersonogren1998 bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.andersonogren1998 input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.andersonogren1998
```

Single instrument

This will correct all scatterings and backscatterings for S11.

```
da.corr.andersonogren1998 --instruments=S11 bnd S11a 2015-05-01 2015-05-03
```

Single variable

This will correct only the variable BbsG_S11.

```
da.corr.andersonogren1998 --backscatterings=bnd:raw:BbsG_S11 bnd S11a 2015-05-01
2015-05-03
```

da.corr.bond1999

This component provides a mechanism to apply the single-angle absorption correction for filter based measurements described in Bond et al. 1999. Data are left at their native wavelengths by applying a 0.97 factor to undo the implicit wavelength adjustment in the original correction.

Usage

da.corr.bond1999 [switches...] [[station] variables times [archive]|[file]]

Switches

--coarse[=BOOLEANS]

When enabled, this sets the K1 values to a function of the scattering Ångström exponent suitable for coarse-mode aerosol.

Default: **Disabled**

--correct=SELECTION

These are the variables to that the correction is applied to. This option is mutually exclusive with instrument specification.

Default: **All absorptions**

--extinction=SELECTION

These are the variables to the light extinction coefficients used by the correction are contained in. By default none are used. This option is mutually exclusive with instrument specification.

--extinction-instruments=SUFFIX...

These are the instrument suffixes that input light extinctions are read from. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

--instruments=SUFFIX...

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--k1=INPUT

This is the fraction of (wavelength adjusted) scattering subtracted from the absorption values. This option is ignored when operating in coarse mode, where the fraction subtracted is a function of the Ångström exponent.

Default: **0.02**

--k2=INPUT

This is the factor that the resulting absorption (after scattering subtraction) is divided by to get the final result. When operating from extinction the K1 value is subtracted from this.

Default: **1.22**

--scattering=SELECTION

These are the variables to the light scattering coefficients used by the correction are contained in. This option is mutually exclusive with instrument specification.

Default: **All scatterings from S11**

--scattering-instruments=SUFFIX...

These are the instrument suffixes that input light scatterings are read from. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **S11**

--smoothing=SMOOTHER

This is the smoothing applied to the scattering and/or extinction inputs before any calculation using them.

Default: **3-minute TC low pass filter**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will correct all absorptions present using the scatterings from the S11 instrument. The default values for K1 (0.02) and K2 (1.22) will be used.

```
da.corr.bond1999 bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.bond1999 input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.bond1999
```

Single instrument with manual constants

This will correct all absorptions from A11 using extinctions from E11. Non-standard constants with K1 = 0.03 and K2 = 1.23 are also used.

```
da.corr.bond1999 --extinction-instruments=E11 --instruments=A11 --k1=0.03 --k2=1.23
bnd S11a 2015-05-01 2015-05-03
```

Single variable in coarse mode

This will correct only the variable BaG_A11 using the coarse mode constants.

```
da.corr.bond1999 --coarse --correct=bnd:raw:BaG_A11 bnd S11a 2015-05-01 2015-05-03
```

da.corr.dilution

This component provides a mechanism to correct data that has been diluted. This results in larger output values, since the dilution air is assumed to be filtered.

Usage

da.corr.dilution [switches...] [[station] variables times [archive]|[file]]

Switches

--correct=**SELECTION**

These are the variables to apply the correction to.

Default: **All counts, scatterings, extinctions, and absorptions**

--dilution=**INPUT**

This is the dilution flow rate to use. The dilution correction factor is equal to the total flow (the sum of all sample and dilution flows) divided by the sum of the sample flows.

--instruments=**SUFFIX...**

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--sample=**INPUT**

This is the sample flow rate to use. The dilution correction factor is equal to the total flow (the sum of all sample and dilution flows) divided by the sum of the sample flows.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular

expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single instrument with manual settings

This will correct S11 parameters using a fixed sample flow of 30 lpm and a fixed dilution flow of 10 lpm.

```
da.corr.dilution --dilution=10 --instruments=S11 --sample=30 bnd S11a 2015-05-01
2015-05-03
```

Or:

```
da.corr.dilution --dilution=10 --instruments=S11 --sample=30 input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.dilution --dilution=10
--instruments=S11 --sample=30
```

da.corr.mueller2011

This component provides a mechanism to apply the Müller et al. (2011) TSI 3563 and Ecotech Aurora 3000 nephelometer truncation correction.

Usage

da.corr.mueller2011 [switches...] [[station] variables times [archive]|[file]]

Switches

--mode=TYPE

This is the correction mode. This determined which set or sets of coefficients to apply. For example, this option can be used to correct data as if it where PM1 (to handle very fine, but not size selected data) or to use the maximum correction factor available (to handle multimodal data).

Default: **Automatic**

The possible values are:

--mode=PM1

Always use PM1 fine mode constants

--mode=PM10

Always use PM10 coarse mode constants

--mode=automatic

Automatically select the constants to use

--mode=automatic

Same as --mode=**automatic**

--mode=coarse

Same as --mode=**PM10**

--mode=fine

Same as --mode=**PM1**

--mode=max

Same as --mode=**maximum**

--mode=maximum

Use the constants that result in the largest correction ratio

--mode=min

Same as --mode=**minimum**

--mode=minimum

Use the constants that result in the smallest correction ratio

--mode=noang

Ignore Ångström exponent fit

--mode=total

Same as --mode=**PM10**

--backscatterings=SELECTION

These are the backscattering variables to correct. This option is mutually exclusive with instrument specification.

Default: **All backscatterings**

--instruments=SUFFIX...

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--scatterings=SELECTION

These are the scattering variables to correct. This option is mutually exclusive with instrument specification.

Default: **All scatterings**

--smoothing=SMOOTHER

This is the smoothing applied to the scatterings before calculation of the Ångström exponent.

Default: **3-minute TC single pole low pass**

--tsi[=BOOLEAN]

When enabled the correction constants used correspond to the TSI 3563 nephelometer instead of the Ecotech Aurora 3000.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will correct all scatterings and backscatterings using the default smoothing. The constants used in the fits and no-angstrom multiplies are those described in Müller et al (2011).

```
da.corr.mueller2011 bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.mueller2011 input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.mueller2011
```

Single instrument

This will correct all scatterings and backscatterings for S11.

```
da.corr.mueller2011 --instruments=S11 bnd S11a 2015-05-01 2015-05-03
```

Single variable

This will correct only the variable BbsG_S11.

```
da.corr.mueller2011 --backscatterings=bnd:raw:BbsG_S11 bnd S11a 2015-05-01 2015-05-
03
```

da.corr.mueller2014

This component provides a mechanism to apply the constrained two-stream radiative transfer model described in Müller 2014. This correction handles both the loading and scattering effects on the filter.

Usage

da.corr.mueller2014 [switches...] [[station] variables times [archive]|[file]]

Switches

--asymmetry-end=**INPUT**

This is the input asymmetry parameter at the end of each sampling interval. That is, the asymmetry parameter calculated from the scattering and back scattering integrated optical depths. When this option is set, this is used for all absorption channels. Therefor, normally only

a single absorption can be corrected while using this option if there is any wavelength dependence.

Default: **Resolved automatically by instrument suffix**

--asymmetry-start=**INPUT**

This is the input asymmetry parameter at the start of each sampling interval. That is, the asymmetry parameter calculated from the scattering and back scattering integrated optical depths. When this option is set, this is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option if there is any wavelength dependence.

Default: **Resolved automatically by instrument suffix**

--correct=**SELECTION**

These are the variables to that the correction is applied to. This option is mutually exclusive with instrument specification.

Default: **All absorptions**

--delta-af=**INPUT**

This is the absorption optical of an unloaded filter used in the model calculation.

Default: **Wavelength dependent**

--delta-sf=**INPUT**

This is the scattering optical of an unloaded filter used in the model calculation.

Default: **Wavelength dependent**

--deltaf-end=**INPUT**

This is the input end filter optical depth used in the correction. When this option is set, this value is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option.

Default: **Resolved automatically by instrument suffix**

--deltaf-start=**INPUT**

This is the input start filter optical depth used in the correction. When this option is set, this value is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option.

Default: **Resolved automatically by instrument suffix**

--deltas-end=**INPUT**

This is the input end scattering optical depth used in the correction. That is, this is the scattering integrated with the sample length at the start of each calculation interval.When this option is set, this is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option if there is any wavelength dependence.

Default: **Resolved automatically by instrument suffix**

--deltas-start=**INPUT**

This is the input start scattering optical depth used in the correction. That is, this is the scattering integrated with the sample length at the start of each calculation interval.When this option is set, this is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option if there is any wavelength dependence.

Default: **Resolved automatically by instrument suffix**

--dl=**INPUT**

This is the change in sample length used for each calculation interval. When omitted it is calculated from the start and end sample lengths.

--gf=**INPUT**

This is the average asymmetry parameter of the filter fibers used in the model calculation.

Default: **0.75**

--hf=**INPUT**

This is the fractional thickness of the particle layer used in the two stream model calculation.

Default: **0.2**

--instruments=**SUFFIX...**

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--length-end=**INPUT**

This is the input end sample length used in the correction. When this option is set, this length is used for all absorption channels.

Default: **Resolved automatically by instrument suffix**

--length-start=**INPUT**

This is the input start sample length used in the correction. When this option is set, this length is used for all absorption channels.

Default: **Resolved automatically by instrument suffix**

--transmittance-end=**INPUT**

This is the input end transmittance used in the correction. When this option is set, this transmittance is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option.

Default: **Resolved automatically by instrument suffix**

--transmittance-start=INPUT

This is the input start transmittance used in the correction. When this option is set, this transmittance is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option.

Default: **Resolved automatically by instrument suffix**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will correct all absorptions present using pre-calculated scattering optical depths associated with each wavelength.

```
da.corr.mueller2014 bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.mueller2014 input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.mueller2014
```

Single instrument with a manual constant

This will correct all absorptions from A11 using an alternate setting for the absorption optical depth of an unloaded filter.

```
da.corr.mueller2014 --delta-af=0.02 --instruments=A11 bnd S11a 2015-05-01 2015-05-
03
```

da.corr.removecontam

This component removes contaminated values from the data stream. It replaces any affected value that occurs during a time when that values source instrument had a contaminated flag set with an undefined value. For example the variable BsG_S11 is removed when the variable F1_S11 has the flag "Contaminated" set. In this case S11 is the instrument and the contamination flags are always present in the F1 variable for an instrument. Any flag starting with "Contaminate" is considered to be a contaminating flag. The set of variables that are affected by contamination can either be specified directly or set base on the standard system definition for a processing stream.

Usage

da.corr.removecontam [switches...] [[station] variables times [archive]|[file]]

Switches

--contamination=STRING

This defines the standard contamination system that is used. The variables affected by contamination are determined by which system is in use. For example, the "aerosol" system affects scatterings, absorptions, extinctions, and counts. This may not be used if variables are specified directly.

Default: **aerosol**

--variables=SELECTION

This defines the variables that are affected by contamination. This may not be used if a system mode is specified.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will remove all variables affected by the standard definition of "aerosol" processing.

```
da.corr.removecontam bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.removecontam input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.removecontam
```

Alternate system mode

This will remove all variables affected by the standard definition of "met" processing.

```
da.corr.removecontam --contamination=met bnd S11a 2015-05-01 2015-05-03
```

Single variable

This will remove only the variable T1_S11 when it is contaminated. All other variables are left unchanged even if they would normally be affected by system contamination.

```
da.corr.removecontam --variables=bnd:raw:T1_S11 bnd S11a 2015-05-01 2015-05-03
```

da.corr.removeundefined

This component provides a mechanism to remove values when (potentially different) values are not defined. The normal use of this is to remove transmittance and length data when absorption are undefined from mentor editing so that difference measurements fragment correctly.

Usage

da.corr.removeundefined [switches...] [[station] variables times [archive]|[file]]

Switches

--instruments=**SUFFIX...**

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--remove=**SELECTION**

These are the variables that are removed when the required variables are not defined.

Default: **Transmittance and length**

--require=**SELECTION**

These are the variables to that must be defined (or absent entirely) for the remove variables to pass through.

Default: **Absorptions**

--undefine=**SELECTION**

These are the variables that set to undefined when the required variables are not defined.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will remove transmittance and length whenever the absorption is not defined.

```
da.corr.removeundefined bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.removeundefined input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.removeundefined
```

Single instrument

This will only remove transmittance and lengths from the A11 instrument whenever any absorption from it is not defined.

```
da.corr.removeundefined --instruments=A11 bnd S11a 2015-05-01 2015-05-03
```

da.corr.script

This component provides a general purpose processor for data using the scripting interface. This processor can be configured to use demultiplexer, value, or segment semantics.

Usage

da.corr.script [switches...] [[station] variables times [archive]|[file]]

Switches

--bypass=SELECTION

This option sets the values that bypass the processor. Bypass values may also be inputs, but should generally not be outputs.

Default: **Anything not input or output**

--code=SCRIPT

This option sets the script code that is evaluated. When in general mode, the incoming data stream is available on the global variable "data". In processor mode, the variable contains the current value and "control" is stream control. In fanout mode the global variable "fanout" contains the fanout controller, which calls the handler with a control stream and fanout key as the two arguments. It expects a return of the handler with an optional output transformer.

--input=SELECTION

This option set the values that are passed into the processor. Only values that this option matches are available as inputs

Default: **Everything**

--mode=TYPE

Set the mode that the processing code is invoked as.

Default: **Processor**

The possible values are:

--mode=fanout

Invoke the processing code as a fanout dispatcher. The fanout controller is available as the global "fanout" and should be called with the control handler before the script finishes.

--mode=general

Invoke the processing code as a general script, expecting it to loop over the incoming data. The global "data" contains the value stream.

--mode=processor

Invoke the processing code repeatedly as a handler. The global "data" contains the current value and "control" contains the stream controller.

--output=SELECTION

This option set the values that are extracted from segments or not ignored by value processors.

Default: **All segment inputs or all values produced**

--type=TYPE

Set the type of data stream provided.

Default: **Segment**

The possible values are:

--type=segment

Provide data sequence segment values, containing multiple variables.

--type=value

Provide individual data values, each representing a single un-flattened variable value.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time

arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Simple operation

This subtracts 0.25 from the BsG_S11 variable. This mode is only recommended for a single set of aligned data, since it forces all data into the same time segments.

```
da.corr.script --code='data.BsG_S11 = data.BsG_S11 - 0.25' bnd S11a 2015-05-01
2015-05-03
```

Or:

```
da.corr.script --code='data.BsG_S11 = data.BsG_S11 - 0.25' input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.script --code='data.BsG_S11 =
data.BsG_S11 - 0.25'
```

Time alteration

This forces data into one minute alignment of starts and ends. Note that this can disrupt some averaging types.

```
da.corr.script --code='control.autorelease = 60; data.START = math.floor(data.START
/ 60) * 60; data.END = math.ceil(data.END / 60) * 60;' --type=value bnd S11a 2015-
05-01 2015-05-03
```

Data removal

This removes all data where the green scattering is less than 0.25.

```
da.corr.script --code='for v in data do if (v.BsG_S11 < 0.25) then data.erase(v);
end end' --mode=general bnd S11a 2015-05-01 2015-05-03
```

Fanout operation

This subtracts 0.25 from BsG_S11 while properly handling overlapping cut sizes, archives, and stations.

```
da.corr.script --code='fanout(function(control) return function(data) data.BsG_S11
= data.BsG_S11 - 0.25; end end)' --mode=fanout bnd S11a 2015-05-01 2015-05-03
```

da.corr.stp

This component provides a mechanism to correct various measurements from ambient conditions to STP.

Usage

da.corr.stp [switches...] [[station] variables times [archive]|[file]]

Switches

--correct-concentration=**SELECTION**

These are the variables to correct that are inversely proportional to the ambient air volume. For example, light scattering. This option is mutually exclusive with instrument specification.

Default: **All counts and extinction components**

--correct-volume=**SELECTION**

These are the variables to correct that are proportional to the ambient air volume. For example, volumetric flows. This option is mutually exclusive with instrument specification.

Default: **All flows**

--instruments=**SUFFIX...**

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--p=**INPUT**

This is the pressure that the measurements where sampled at in hPa. The result is the correction of the data from this pressure STP.

Default: **Resolved automatically by instrument suffix**

--stp-p=**INPUT**

This is the pressure that the input is corrected to in hPa.

Default: **1013.25 hPa**

--stp-t=INPUT

This is the temperature that the input is corrected to. This is assumed to be in kelvin if it is greater than 150, otherwise it is in degrees celsius.

Default: **273.15 K**

--t=INPUT

This is the temperature that the measurements where sampled at. The result is the correction of the data from this temperature STP. This is assumed to be in kelvin if it is greater than 150, otherwise it is in degrees celsius.

Default: **Resolved automatically by instrument suffix**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data

are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will correct all counts, flow rates, and light scattering, absorption, or extinction values present in the data stream. The values are corrected using a temperature and pressure from the same suffix (e.x. BsG_S11 uses T_S11 and P_S11). They are corrected to 273.15 K and 1013.25 hPa.

```
da.corr.stp bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.stp input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.stp
```

Single instrument with manual pressure

This will correct all parameters for S11 to its temperature (from T_S11 for example) and 880.0 hPa.

```
da.corr.stp --instruments=S11 --p=880 bnd S11a 2015-05-01 2015-05-03
```

Single variable and alternate temperature

This will correct only the variable BsG_S11 and specifies an alternate sample temperature (Tu_S11).

```
da.corr.stp --correct-concentration=bnd:raw:BsG_S11 --t=bnd:raw:Tu_S11 bnd S11a
2015-05-01 2015-05-03
```

da.corr.virkkula2005

This component provides a mechanism to apply the single scattering albedo dependent PSAP correction described in Virkkula et all 2005. This correction incorporates the updated constants from the 2010 errata. The correction involved a loading correction dependent on single scattering albedo and a constant fraction of scattering subtraction.

Usage

da.corr.virkkula2005 [switches...] [[station] variables times [archive]|[file]]

Switches

--correct=**SELECTION**

These are the variables to that the correction is applied to. This option is mutually exclusive with instrument specification.

Default: **All absorptions**

--extinction=**SELECTION**

These are the variables to the light extinction coefficients used by the correction are contained in. By default none are used. This option is mutually exclusive with instrument specification.

--extinction-instruments=**SUFFIX...**

These are the instrument suffixes that input light extinctions are read from. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

--h0=**INPUT**

This is the h1 constant used to generate the absorption scaling factor. Ba = [k0 + k1×(h0 + h1×ω)×ln(Tr)]×σa0 - s×σs

Default: **Wavelength dependent**

--h1=**INPUT**

This is the h1 constant used to generate the absorption scaling factor. Ba = [k0 + k1×(h0 + h1×ω)×ln(Tr)]×σa0 - s×σs

Default: **Wavelength dependent**

--instruments=**SUFFIX...**

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--k0=**INPUT**

This is the k0 constant used to generate the absorption scaling factor. Ba = [k0 + k1×(h0 + h1×ω)×ln(Tr)]×σa0 - s×σs

Default: **Wavelength dependent**

--k1=**INPUT**

This is the k1 constant used to generate the absorption scaling factor. Ba = [k0 + k1×(h0 + h1×ω)×ln(Tr)]×σa0 - s×σs

Default: **Wavelength dependent**

--s=**INPUT**

This is the s constant that is multiplied by the scattering and subtracted from the absorption. Ba = [k0 + k1×(h0 + h1×ω)×ln(Tr)]×σa0 - s×σs

Default: **Wavelength dependent**

--scattering=**SELECTION**

These are the variables to the light scattering coefficients used by the correction are contained in. This option is mutually exclusive with instrument specification.

Default: **All scatterings from S11**

--scattering-instruments=**SUFFIX...**

These are the instrument suffixes that input light scatterings are read from. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **S11**

--smoothing=**SMOOTHER**

This is the smoothing applied to the scattering and/or extinction inputs before any calculation using them.

Default: **3-minute TC low pass filter**

--transmittance-end=**INPUT**

This is the input end transmittance used in the correction. When this option is set, this transmittance is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option.

Default: **Resolved automatically by instrument suffix**

--transmittance-start=**INPUT**

This is the input start transmittance used in the correction. When this option is set, this transmittance is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option.

Default: **Resolved automatically by instrument suffix**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will correct all absorptions present using the scatterings from the S11 instrument.

```
da.corr.virkkula2005 bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.virkkula2005 input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.virkkula2005
```

Single instrument with manual constants

This will correct all absorptions from A11 using extinctions from E11. Non-standard constants with k0 = 0.534 and k1 = -0.617 are also used.

```
da.corr.virkkula2005 --extinction-instruments=E11 --instruments=A11 --k0=0.534
--k1=-0.617 bnd S11a 2015-05-01 2015-05-03
```

Single variable

This will correct only the variable BaG_A11.

```
da.corr.virkkula2005 --correct=bnd:raw:BaG_A11 bnd S11a 2015-05-01 2015-05-03
```

da.corr.wavelength

This component provides a mechanism to adjust input data to different wavelengths via interpolation.

Usage

da.corr.wavelength [switches...] [[station] variables times [archive]|[file]]

Switches

--adjust=SELECTION

These are the variables to apply the adjustment to. They are matched to the nearest wavelength in the target set.

Default: **All scatterings, extinctions, and absorptions**

--angstrom=**INPUT**

This is the Ångström exponent used to perform wavelength adjustment when interpolation is not possible. When not set, data are simply left as undefined when they cannot be adjusted to the target wavelengths.

--angstrom-distance=**NUMBERS**

This is the maximum distance that the fallback Ångström exponent is valid for. This is used in conjunction with angstrom-wavelength to limit how far data are extrapolated.

This option only accepts numbers greater than or equal to 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--angstrom-wavelength=**NUMBERS**

This is the wavelength of the Ångström exponent used to perform wavelength adjustment when interpolation is not possible. This is used to limit how far data are extrapolated when used in conjunction with angstrom-distance.

This option only accepts numbers strictly greater than 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--create-outputs[=**BOOLEAN**]

When enabled, this will cause the creation of outputs at the specified wavelengths, even if a matching one does not exist in the input.

--instruments=**SUFFIX...**

These are the instrument suffixes to correct. For example S11 would usually specifies the reference nephelometer. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--smoothing=**SMOOTHER**

This is the smoothing applied to inputs before adjustment.

--wavelengths=**NUMBER...**

These are the wavelengths that the input data are adjusted to. Input data are adjusted to the nearest matching wavelength, if there are more data than wavelengths the further ones are left un-adjusted.

Default: **450,550,700**

This option only accepts numbers strictly greater than 0.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code

of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will correct light scattering, absorption and extinctions present in the input to 450, 550, and 700 nm, using whichever is nearest to the original wavelength.

```
da.corr.wavelength bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.wavelength input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.wavelength
```

Single instrument with manual wavelength

This will correct the nearest wavelength from S11 to 567 nm.

```
da.corr.wavelength --instruments=S11 --wavelengths=567 bnd S11a 2015-05-01 2015-05-
03
```

da.corr.weiss

This component applies the loading correction implemented by Ray Weiss for the Radiance Research PSAP. This correction consists of dividing the absorptions by a polynomial of the corresponding transmittance. Under normal circumstances, this polynomial is just a slope and intercept. That is, the normal form of the correction is: out = in / (A + B * Tr)

Usage

da.corr.weiss [switches...] [[station] variables times [archive]|[file]]

Switches

--constants=COEFFICIENTS...

This calibration set the transformation applied to the transmittances to generate the correction factor.

Default: **0.814,1.237**

--correct-absorption=SELECTION

These are the variables that the correction is applied to. Correcting using more than one transmittance for all absorptions is only supported using the suffixes option or by leaving this option unset (using all suffixes).

Default: **All absorptions**

--instruments=SUFFIX...

These are the instrument suffixes to correct. For example A11 would usually specifies the main absorption instrument. This option is mutually exclusive with manual variable specification.

Default: **All instrument suffixes**

--reverse[=BOOLEANS]

When enabled, this causes the correction to be run in reverse. This is used to back out the correction as applied by the PSAPs in their hardware.

--transmittance-end=INPUT

This is the input end transmittance used in the correction. When this option is set, this transmittance is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option.

Default: **Resolved automatically by instrument suffix**

--transmittance-start=INPUT

This is the input start transmittance used in the correction. When this option is set, this transmittance is used for all absorption channels. Therefor, normally only a single absorption can be corrected while using this option.

Default: **Resolved automatically by instrument suffix**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time

arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will correct all absorptions present in the input data. The transmittance for each absorption is resolved by looking for one with the same end suffix as the absorption. For example, BaG_A11 uses IrG_A11 as its transmittance.

```
da.corr.weiss bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.corr.weiss input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.corr.weiss
```

Single instrument with alternate constants

This will correct all absorptions from the A11 instrument using the alternate constants A=0.866 and B=1.317.

```
da.corr.weiss --constants=0.866:1.317 --instruments=A11 bnd S11a 2015-05-01 2015-
05-03
```

Single variable with alternate transmittance

This will correct only the variable BaG_A11 and specifies an alternate transmittance (IrR_A11) to correct it with.

```
da.corr.weiss --correct-absorption=bnd:raw:BaG_A11 --transmittance
-start=bnd:raw:IrR_A11 bnd S11a 2015-05-01 2015-05-03
```

da.smooth.1plp

This applies a single pole low pass digital filter to all data. This time constant is compensated for variable intervals between values. Note that some types of averaging require inputs beyond the values being averaged. For example, to average raw absorptions, transmittance and path length are required because the average is performed with a difference measurement.

Usage

da.smooth.1plp [switches...] [[station] variables times [archive]|[file]]

Switches

--gap=INTERVAL

If two values are seperated by this much time then the smoother is reset. That is, data separated by this much will be smoothed as independent runs of data.

Default: **Infinite**

Undefined intervals are accepted. The interval must be greater than or equal to zero length.

--ignore-undefined[=BOOLEAN**]**

If this option is set undefined values are ignored instead of forcing a smoother reset.

Default: **Disabled, undefined values cause a reset**

--tc=INTERVAL

This is the time constant of the filter. A time constant for a digital filter is the time required for a step change in values to reach 63.2% of the final value.

Default: **Three minutes**

The interval must be greater than zero in length.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Default

This smooths all available data with a single pole low pass digital filter with a time constant of three minutes. Gaps of any size are smoothed over (taken as continuous sequences of data) but undefined values cause the smoother to reset.

```
da.smooth.1plp bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.smooth.1plp input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.smooth.1plp
```

Alternate time constant

This smooths data with a digital filter with a time constant of one hour.

```
da.smooth.1plp --tc=1h bnd S11a 2015-05-01 2015-05-03
```

da.smooth.3rssh

This applies a Tukey 3RSSH smoother to all data. This does NOT compensate variable time interval data. That is each variable smoothed is assumed to be present at uniform intervals. Note that some types of averaging require inputs beyond the values being averaged. For example, to average raw absorptions, transmittance and path length are required because the average is performed with a difference measurement.

Usage

da.smooth.3rssh [switches...] [[station] variables times [archive]|[file]]

Switches

--gap=**INTERVAL**

If two values are seperated by this much time then the smoother is reset. That is, data separated by this much will be smoothed as independent runs of data.

Default: **Infinite**

Undefined intervals are accepted. The interval must be greater than or equal to zero length.

--ignore-undefined[=**BOOLEAN]**

If this option is set undefined values are ignored instead of forcing a smoother reset.

Default: **Disabled, undefined values cause a reset**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code

of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Default

This smooths all available data with a Tukey 3RSSH smoother. Gaps of any size are smoothed over (taken as continuous sequences of data) but undefined values cause the smoother to reset.

```
da.smooth.3rssh bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.smooth.3rssh input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.smooth.3rssh
```

da.smooth.4plp

This applies a four pole low pass digital filter to all data. This time constant is compensated for variable intervals between values. Note that some types of averaging require inputs beyond the values being averaged. For example, to average raw absorptions, transmittance and path length are required because the average is performed with a difference measurement.

Usage

da.smooth.4plp [switches...] [[station] variables times [archive]|[file]]

Switches

--gap=**INTERVAL**

If two values are seperated by this much time then the smoother is reset. That is, data separated by this much will be smoothed as independent runs of data.

Default: **Infinite**

Undefined intervals are accepted. The interval must be greater than or equal to zero length.

--ignore-undefined[=**BOOLEAN]**

If this option is set undefined values are ignored instead of forcing a smoother reset.

Default: **Disabled, undefined values cause a reset**

--tc=**INTERVAL**

This is the time constant of the filter. A time constant for a digital filter is the time required for a step change in values to reach 63.2% of the final value.

Default: **Three minutes**

The interval must be greater than zero in length.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Default

This smooths all available data with a single pole low pass digital filter with a time constant of three minutes. Gaps of any size are smoothed over (taken as continuous sequences of data) but undefined values cause the smoother to reset.

```
da.smooth.4plp bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.smooth.4plp input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.smooth.4plp
```

Alternate time constant

This smooths data with a digital filter with a time constant of one hour.

```
da.smooth.4plp --tc=1h bnd S11a 2015-05-01 2015-05-03
```

da.smooth.df

This applies a manually specified digital filter to the input data. The specification is a set of coefficients applied to the history of input and smoothed values. Note that some types of averaging require inputs beyond the values being averaged. For example, to average raw absorptions, transmittance and path length are required because the average is performed with a difference measurement.

Usage

da.smooth.df [switches...] [[station] variables times [archive]|[file]]

Switches

--a=NUMBER...

These are the constants multiplied with the un-smoothed values. during filter calculation. The more recent ones are first (leftmost) in the list. For example the multiplier on the value being smoothed is the first value on the left.

Default: **1.0**

This option requires at least one number if it is used.

--b=NUMBER...

These are the constants multiplied with the smoothed values. during filter calculation. The more recent ones are first (leftmost) in the list. For example the multiplier on the previous output is the first value on the left.

This option may also be set to an empty value to indicate an empty list of numbers.

--gap=INTERVAL

If two values are seperated by this much time then the smoother is reset. That is, data separated by this much will be smoothed as independent runs of data.

Default: **Infinite**

Undefined intervals are accepted. The interval must be greater than or equal to zero length.

--ignore-undefined[=BOOLEAN]

If this option is set undefined values are ignored instead of forcing a smoother reset.

Default: **Disabled, undefined values cause a reset**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an

archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single pole low pass

This filter specifies a single pole low pass configuration with a time constant of five data points. This will smooth all input data over any gap length but will reset on undefined values.

```
da.smooth.df --a=0.18127 --b=0.81873 bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.smooth.df --a=0.18127 --b=0.81873 input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.smooth.df --a=0.18127 --b=0.81873
```

da.smooth.fourier

This applies a smoother based on zeroing parts of the frequency spectrum generated from a Fourier transform. This allows for construction of a low pass, high pass, band pass or band stop filter by specifying the various parts to zero out. Note that some types of averaging require inputs beyond the values being averaged. For example, to average raw absorptions, transmittance and path length are required because the average is performed with a difference measurement.

Usage

da.smooth.fourier [switches...] [[station] variables times [archive]|[file]]

Switches

--gap=**INTERVAL**

If two values are seperated by this much time then the smoother is reset. That is, data separated by this much will be smoothed as independent runs of data.

Default: **Infinite**

Undefined intervals are accepted. The interval must be greater than or equal to zero length.

--high=**INTERVAL**

This the high cutoff period. All frequencies with periods longer than this are removed. However when used with a low cutoff that would be mutually exclusive (longer) it is instead interpreted to be the lower bound of a band stop filter. In this context is specifies the shortest period frequency to be removed.

Undefined intervals are accepted. The interval must be greater than zero in length.

--ignore-undefined[=**BOOLEAN**]

If this option is set undefined values are ignored instead of forcing a smoother reset and another Fourier transform.

Default: **Enabled, undefined values are ignored**

--low=**INTERVAL**

This is the low period cutoff. All frequencies with periods shorter than this are removed. However when used with a high cutoff that would be mutually exclusive (shorter) it is instead interpreted to be the upper bound of a band stop filter. In this case is specifies the longest period frequency to be removed.

Undefined intervals are accepted. The interval must be greater than zero in length.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Low pass filter

This create a low pass filter by removing all frequencies below one hour in period.

```
da.smooth.fourier --low=1h bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.smooth.fourier --low=1h input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.smooth.fourier --low=1h
```

Band stop filter

This create a band stop filter that removes all frequencies between 50 and 70 minutes in period. This will nominally remove any hourly cycles in the data.

```
da.smooth.fourier --high=50m --low=70m bnd S11a 2015-05-01 2015-05-03
```

da.smooth.segment

This component applies conventional averaging to bins defined by part of its input. That is, it averages high resolution data to the times a lower resolution variable or variables exist. Normally it also removes contaminated values from the averages. This can, for example, be used to generate whole filter PSAP/CLAP averages. Note that some types of averaging require inputs beyond the values being averaged. For example, to average raw absorptions, transmittance and path length are required because the average is performed with a difference measurement.

Usage

da.smooth.segment [switches...] [[station] variables times [archive]|[file]]

Switches

--contamination=**STRING**

This defines the contamination mode in effect. The contamination mode determines which variables are removed when they are flagged as contaminated. For example the "aerosol" mode generally removes scattering, absorption, extinction and concentrations. The special mode "none" or "disable" turns off all contamination removal and values are averaged regardless of their contaminated state.

Default: **aerosol**

--continuous[=**BOOLEAN]**

If this option is set then the averages produced are the maximum continuous averages within the binning interval. This allows difference measurements to be repeatable but may cause splitting within the interval.

Default: **Disabled**

--cover=**NUMBERS**

This is the fraction of data required to exist for an average bin to be produced at all. For example if this is set to 0.95 then averages will only be produced when the fraction of missing data in a bin is greater than 95% of the total bin. So, for a one hour bin that would require at least 57 minutes of valid data. Setting this to undefined disables the requirement.

Default: **Disabled**

--discard-intersecting[=**BOOLEAN]**

If this option is set then all input data that is not completely contained within an averaging segment is discard and does not contribute to that segment. That is, any value that overlaps the start or end of the segment will not be used.

Default: **Disabled; include intersecting data**

--gap=**INTERVAL**

If two values are seperated by this much time then the average is split. Even if there is no gap the values may still be split into separate bins if the interval demands it. An undefined gap

allows for infinite separation.

Default: **Infinite**

Undefined intervals are accepted. The interval must be greater than or equal to zero length.

--output=**SELECTION**

If this is set then the final output segments are written out as the variable or variables that this defines.

--segment=**SELECTION**

This defines the data input to create segment averages on. The resulting averages are output for any times that this exists. If there are overlaps they are fragmented into multiple segment averages.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Default zero averages

This will average all data available during the neph zero. It will also remove contamination from the standard set of variables involved in an aerosol system (generally scattering, absorption, extinction and concentrations).

```
da.smooth.segment --segment=bnd:raw:ZZero_S11 bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.smooth.segment --segment=bnd:raw:ZZero_S11 input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.smooth.segment
--segment=bnd:raw:ZZero_S11
```

Mode and coverage requirement

This will average data to whenever the neph has changed operating modes while requiring at least 95% of data within each segment to exist. Contaminated data is not excluded from the averages.

```
da.smooth.segment --contamination=none --cover=0.95 --segment=bnd:raw:F2_S11 bnd
S11a 2015-05-01 2015-05-03
```

Plotting

These programs generate graphical representations of data.

da.allan

A plot of standard deviation versus averaging time.

Usage

da.allan [switches...] [[station] [variables] times [archive]|[file]]

Switches

--file=**FILE**

This is the output image file to write to. Supported formats: bmp, cur, ico, jpeg, jpg, pbm, pgm, png, ppm, svg, xbm, xpm.

Default: **output.png**

--fit[=**BOOLEAN]**

This controls if the fit line is enabled.

Default: **Enabled**

--fit-end=**NUMBER**

The end bound of the data that contributes to the fit. Any data after this will be ignored in the fit.

Default: **100 seconds**

This option only accepts numbers strictly greater than 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--fit-start=**NUMBER**

The start bound of the data that contributes to the fit. Any data before this will be ignored in the fit.

Default: **10 seconds**

This option only accepts numbers strictly greater than 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--grid=**TYPE**

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--height=INTEGER

This is the height of the output file.

Default: **768**

This option only accepts integers greater than or equal to 1.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top right corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--opaque[=BOOLEAN]

When enabled on an output format that supports transparency, this causes the output image to have an opaque white background instead of the normal transparent one.

--points[=BOOLEAN]

This enables the display of the discrete points.

Default: **Disabled**

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=STRING

This is the main graph title.

--width=INTEGER

This is the width of the output file.

Default: **1024**

This option only accepts integers greater than or equal to 1.

--x-title=STRING

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=VARIABLE

This is the variable or variables to the standard deviation of on the Y axis.

--y-title=STRING

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=VARIABLE

This is the variable or variables that determine the trace color. This generates a continuum of colors between minimum and maximum values for the variable.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single trace

This will create a simple Allan plot of PM10 data for BaG_A11.

```
da.plot.allan --y sfa A11a 2012-12-03T02:13:44Z 2012-12-03T12:00:00Z
```

da.cdf

A cumulative distribution plot.

Usage

da.cdf [switches...] [[station] [variables] times [archive]|[file]]

Switches

--file=FILE

This is the output image file to write to. Supported formats: bmp, cur, ico, jpeg, jpg, pbm, pgm, png, ppm, svg, xbm, xpm.

Default: **output.png**

--fit[=BOOLEAN]

This sets the display of the distribution fits.

Default: **Enabled**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both
: Both X and Y grid lines are displayed.

--grid=disabled
: No grid is displayed.

--grid=enable
: Same as --grid=**both**

--grid=enabled
: Same as --grid=**both**

--grid=none
: Same as --grid=**disabled**

--grid=off
: Same as --grid=**disabled**

--grid=x
: Only grid lines extending from the X axis are displayed.

--grid=x-y
: Same as --grid=**both**

--grid=xy
: Same as --grid=**both**

--grid=y
: Only grid lines extending from the Y axis are displayed.

--height=**INTEGER**
: This is the height of the output file.

Default: **768**

This option only accepts integers greater than or equal to 1.

--legend=**TYPE**
: This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN]

This changes the variable axes to be logarithmic.

--opaque[=BOOLEAN]

When enabled on an output format that supports transparency, this causes the output image to have an opaque white background instead of the normal transparent one.

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=STRING

This is the main graph title.

--width=**INTEGER**

This is the width of the output file.

Default: **1024**

This option only accepts integers greater than or equal to 1.

--x=**VARIABLE**

This is the variable or variables to plot on the X axis.

--x-title=**STRING**

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y-title=**STRING**

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=**VARIABLE**

This is the variable or variables that determine the trace color. This generates a continuum of colors between minimum and maximum values for the variable.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables

defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Simple

This will create a basic CDF plot.

```
da.plot.cdf --log --x bnd A11a 2015-05-01 2015-05-03
```

da.cycle

A box-whisker plot of periodic cycles.

Usage

da.cycle [switches...] [[station] [variables] times [archive]|[file]]

Switches

--box[=BOOLEAN]

When disabled this hides the box and displays only the whiskers and median.

Default: **Enabled**

--division=INTERVALBLOCK

The X axis is divided based on this interval. For example, if this is set to one month then a

division and bin appear for every month in the period. This cannot be greater than the total period. Alignment is ignored (implicitly aligned with the period).

Default: **One month**

When no units are specified the interval is assumed to be in months.

--extend[=BOOLEAN]

If set then the median lines of the bins are extended to cover the entire graph. If totals are enabled only the median of the totals is extended.

Default: **Enabled with a total bin**

--file=FILE

This is the output image file to write to. Supported formats: bmp, cur, ico, jpeg, jpg, pbm, pgm, png, ppm, svg, xbm, xpm.

Default: **output.png**

--fit=TYPE

This is the type of fit displayed on the graph.

Default: **Disabled**

The possible values are:

--fit=2poly

Same as --fit=**secondorder**

--fit=3poly

Same as --fit=**thirdorder**

--fit=disabled

No fit is displayed.

--fit=leastsquares

Same as --fit=**linear**

--fit=linear

Linear least squares.

--fit=linearzero

Linear least squares forced through zero.

--fit=lsq

Same as --fit=**linear**

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--fit=secondorder

Second order polynomial.

--fit=secondorderzero

Second order polynomial forced through zero.

--fit=thirdorder

Third order polynomial.

--fit=thirdorderzero

Third order polynomial forced through zero.

--fit=zerolinear

Same as --fit=**linearzero**

--fit=zerosecondorder

Same as --fit=**secondorderzero**

--fit=zerothirdorder

Same as --fit=**thirdorderzero**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--height=**INTEGER**

This is the height of the output file.

Default: **768**

This option only accepts integers greater than or equal to 1.

--legend=**TYPE**

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN]

This changes the variable axes to be logarithmic.

--opaque[=BOOLEAN]

When enabled on an output format that supports transparency, this causes the output image to have an opaque white background instead of the normal transparent one.

--period=INTERVALBLOCK

This is the total period of the display. Data is plotted such that the X axis repeats on this period.

Default: **One year**

When no units are specified the interval is assumed to be in years.

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=STRING

This is the main graph title.

--total[=BOOLEAN]

This enables the display of a bin that contains all data.

--width=INTEGER

This is the width of the output file.

Default: **1024**

This option only accepts integers greater than or equal to 1.

--x-title=STRING

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=VARIABLE

This is the variable or variables to plot on the Y axis.

--y-title=STRING

This is the title output along the Y axis.

Default: **Generated from the axis units.**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

Examples

Hourly

This will create a daily cycle plot with bins for each hour of the day.

```
da.plot.cycle --division --period --y bnd A11a 2015-05-01 2015-05-03
```

da.density

A two dimensional scatter plot of point density.

Usage

da.density [switches...] [[station] [variables] times [archive]|[file]]

Switches

--density-limit=NUMBER

The limit of the contribution of any point to the total density as a percentage of the total number of points.

Default: **0.5% of data**

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--file=FILE

This is the output image file to write to. Supported formats: bmp, cur, ico, jpeg, jpg, pbm, pgm, png, ppm, svg, xbm, xpm.

Default: **output.png**

--fit=TYPE

This is the type of fit displayed on the graph.

Default: **Disabled**

The possible values are:

--fit=2poly

Same as --fit=**secondorder**

--fit=3poly

Same as --fit=**thirdorder**

--fit=disabled

No fit is displayed.

--fit=leastsquares

Same as --fit=**linear**

--fit=linear

Linear least squares.

--fit=linearzero

Linear least squares forced through zero.

--fit=lsq

Same as --fit=**linear**

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--fit=secondorder

Second order polynomial.

--fit=secondorderzero

Second order polynomial forced through zero.

--fit=thirdorder

Third order polynomial.

--fit=thirdorderzero

Third order polynomial forced through zero.

--fit=zerolinear

Same as --fit=**linearzero**

--fit=zerosecondorder

Same as --fit=**secondorderzero**

--fit=zerothirdorder

Same as --fit=**thirdorderzero**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both
Both X and Y grid lines are displayed.

--grid=disabled
No grid is displayed.

--grid=enable
Same as --grid=**both**

--grid=enabled
Same as --grid=**both**

--grid=none
Same as --grid=**disabled**

--grid=off
Same as --grid=**disabled**

--grid=x
Only grid lines extending from the X axis are displayed.

--grid=x-y
Same as --grid=**both**

--grid=xy
Same as --grid=**both**

--grid=y
Only grid lines extending from the Y axis are displayed.

--height=INTEGER
This is the height of the output file.

Default: **768**

This option only accepts integers greater than or equal to 1.

--legend=TYPE
This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN]

This changes the variable axes to be logarithmic.

--opaque[=BOOLEAN]

When enabled on an output format that supports transparency, this causes the output image to have an opaque white background instead of the normal transparent one.

--points[=BOOLEAN]

This enables the display of the discrete points.

Default: **Disabled**

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined

values are permitted, to indicate that the option is set but not to a specific value.

--title=**STRING**

This is the main graph title.

--width=**INTEGER**

This is the width of the output file.

Default: **1024**

This option only accepts integers greater than or equal to 1.

--x=**VARIABLE**

This is the variable or variables to plot on the X axis.

--x-title=**STRING**

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=**VARIABLE**

This is the variable or variables to plot on the Y axis.

--y-title=**STRING**

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=**VARIABLE**

This is the variable or variables that determine the trace color. This generates a continuum of colors between minimum and maximum values for the variable. For example, it could be used to display the humidity for points comparing the scattering between two nephelometers.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are

explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single trace

This will create a simple density plot of BaG_A12 vs BaG_A11 for PM10 data.

```
da.plot.density --x --y bnd BaG_A11,BaG_A12 2015-05-01 2015-05-03
```

da.layout

An already defined layout of other plots.

Usage

da.layout [switches...] [[station] [variables] times [archive]|[file]]

Switches

--configuration=SELECTION

The variable in the archive used to load the layout.

--file=FILE

This is the output image file to write to. Supported formats: bmp, cur, ico, jpeg, jpg, pbm, pgm, png, ppm, svg, xbm, xpm.

Default: **output.png**

--height=INTEGER

This is the height of the output file.

Default: **768**

This option only accepts integers greater than or equal to 1.

--input=FILE

If set then the layout is loaded from this file.

--opaque[=BOOLEAN**]**

When enabled on an output format that supports transparency, this causes the output image to have an opaque white background instead of the normal transparent one.

--path=STRING

The path within the configuration to use.

--width=INTEGER

This is the width of the output file.

Default: **1024**

This option only accepts integers greater than or equal to 1.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or

more variable specifications separated by , (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply BsB_S11 selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead ::BsB_S11:pm10 is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as bnd:BsB_S11,thd:BsB_S11 allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value everything can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as raw or clean_meta. The argument accepts multiple archives specified as regular expressions and separated by : or ; or ,. For example raw(_meta)? selects both the raw and raw_meta archives.

The special value allarchives may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

Examples

Static file

This will load the layout contained in the file "layout.xml".

```
da.plot.layout --input=layout.xml bnd S11a 2015-05-01 2015-05-03
```

Archive display

This a display from the main system configuration.

```
da.plot.layout --path=Templates/Graphs/Scattering bnd S11a 2015-05-01 2015-05-03
```

da.pdf

A probability distribution plot.

Usage

da.pdf [switches...] [[station] [variables] times [archive]|[file]]

Switches

--bins=INTEGER

This is the number of bins spread uniformly across the data range used in the histogram.

Default: **10**

This option only accepts integers greater than or equal to 2 and less than or equal to 10000.

--file=FILE

This is the output image file to write to. Supported formats: bmp, cur, ico, jpeg, jpg, pbm, pgm, png, ppm, svg, xbm, xpm.

Default: **output.png**

--fit=TYPE

This is the type of fit used on the histogram.

Default: **Cublic spline**

The possible values are:

--fit=cspline

Cubic spline interpolation.

--fit=cubicspline

Same as --fit=**cspline**

--fit=disabled

No fit is displayed.

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both
: Both X and Y grid lines are displayed.

--grid=disabled
: No grid is displayed.

--grid=enable
: Same as --grid=**both**

--grid=enabled
: Same as --grid=**both**

--grid=none
: Same as --grid=**disabled**

--grid=off
: Same as --grid=**disabled**

--grid=x
: Only grid lines extending from the X axis are displayed.

--grid=x-y
: Same as --grid=**both**

--grid=xy
: Same as --grid=**both**

--grid=y
: Only grid lines extending from the Y axis are displayed.

--height=INTEGER
: This is the height of the output file.

Default: **768**

This option only accepts integers greater than or equal to 1.

--legend=TYPE
: This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft
: The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN]

This changes the variable axes to be logarithmic.

--opaque[=BOOLEAN]

When enabled on an output format that supports transparency, this causes the output image to have an opaque white background instead of the normal transparent one.

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--show-bins[=BOOLEAN]

This enables the display of the histogram bins used in the PDF.

--title=STRING

This is the main graph title.

--width=**INTEGER**

This is the width of the output file.

Default: **1024**

This option only accepts integers greater than or equal to 1.

--x=**VARIABLE**

This is the variable or variables to plot on the X axis.

--x-title=**STRING**

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y-title=**STRING**

This is the title output along the Y axis.

Default: **Generated from the axis units.**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Simple

This will create a basic PDF plot.

```
da.plot.pdf --log --x bnd A11a 2015-05-01 2015-05-03
```

da.scatter

A two dimensional scatter plot of values.

Usage

da.scatter [switches...] [[station] [variables] times [archive]|[file]]

Switches

--file=FILE

This is the output image file to write to. Supported formats: bmp, cur, ico, jpeg, jpg, pbm, pgm, png, ppm, svg, xbm, xpm.

Default: **output.png**

--fit=TYPE

This is the type of fit displayed on the graph.

Default: **Disabled**

The possible values are:

--fit=2poly

Same as --fit=**secondorder**

--fit=3poly

Same as --fit=**thirdorder**

--fit=disabled

No fit is displayed.

--fit=leastsquares

Same as --fit=**linear**

--fit=linear

Linear least squares.

--fit=linearzero

Linear least squares forced through zero.

--fit=lsq

Same as --fit=**linear**

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--fit=secondorder

Second order polynomial.

--fit=secondorderzero

Second order polynomial forced through zero.

--fit=thirdorder

Third order polynomial.

--fit=thirdorderzero

Third order polynomial forced through zero.

--fit=zerolinear

Same as --fit=**linearzero**

--fit=zerosecondorder

Same as --fit=**secondorderzero**

--fit=zerothirdorder

Same as --fit=**thirdorderzero**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--height=INTEGER

This is the height of the output file.

Default: **768**

This option only accepts integers greater than or equal to 1.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft
: The legend is located inside the trace area at the bottom left corner.

--legend=bottomright
: The legend is located inside the trace area at the bottom right corner.

--legend=disabled
: No legend is displayed.

--legend=inside
: Same as --legend=**topleft**

--legend=left
: The legend is located outside and to the left of the graph.

--legend=none
: Same as --legend=**disabled**

--legend=off
: Same as --legend=**disabled**

--legend=outside
: Same as --legend=**right**

--legend=right
: The legend is located outside and to the right of the graph.

--legend=topleft
: The legend is located inside the trace area at the top left corner.

--legend=topright
: The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN]
: This changes the variable axes to be logarithmic.

--opaque[=BOOLEAN]
: When enabled on an output format that supports transparency, this causes the output image to have an opaque white background instead of the normal transparent one.

--quantile=NUMBER
: This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

  This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=**STRING**

This is the main graph title.

--width=**INTEGER**

This is the width of the output file.

Default: **1024**

This option only accepts integers greater than or equal to 1.

--x=**VARIABLE**

This is the variable or variables to plot on the X axis.

--x-title=**STRING**

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=**VARIABLE**

This is the variable or variables to plot on the Y axis.

--y-title=**STRING**

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=**VARIABLE**

This is the variable or variables that determine the trace color. This generates a continuum of colors between minimum and maximum values for the variable. For example, it could be used to display the humidity for points comparing the scattering between two nephelometers.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut

sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single trace

This will create a simple scatter plot of BaG_A12 vs BaG_A11 for all cut sizes.

```
da.plot.scatter --x --y bnd BaG_A11,BaG_A12 2015-05-01 2015-05-03
```

da.timeseries

A two dimensional time series of values.

Usage

da.timeseries [switches...] [[station] [variables] times [archive]|[file]]

Switches

--file=FILE

This is the output image file to write to. Supported formats: bmp, cur, ico, jpeg, jpg, pbm, pgm, png, ppm, svg, xbm, xpm.

Default: **output.png**

--fit=TYPE

This is the type of fit displayed on the graph.

Default: **Disabled**

The possible values are:

--fit=2poly

Same as --fit=**secondorder**

--fit=3poly

Same as --fit=**thirdorder**

--fit=disabled

No fit is displayed.

--fit=leastsquares

Same as --fit=**linear**

--fit=linear

Linear least squares.

--fit=linearzero

Linear least squares forced through zero.

--fit=lsq

Same as --fit=**linear**

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--fit=secondorder

Second order polynomial.

--fit=secondorderzero

Second order polynomial forced through zero.

--fit=thirdorder

Third order polynomial.

--fit=thirdorderzero

Third order polynomial forced through zero.

--fit=zerolinear

Same as --fit=**linearzero**

--fit=zerosecondorder

Same as --fit=**secondorderzero**

--fit=zerothirdorder

Same as --fit=**thirdorderzero**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--height=INTEGER

This is the height of the output file.

Default: **768**

This option only accepts integers greater than or equal to 1.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN**]**

This changes the variable axes to be logarithmic.

--opaque[=BOOLEAN]

When enabled on an output format that supports transparency, this causes the output image to have an opaque white background instead of the normal transparent one.

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=STRING

This is the main graph title.

--width=INTEGER

This is the width of the output file.

Default: **1024**

This option only accepts integers greater than or equal to 1.

--x-title=STRING

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=VARIABLE

This is the variable or variables to plot on the Y axis.

--y-title=STRING

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=VARIABLE

This is the variable or variables that determine the trace color. This generates a continuum of colors between minimum and maximum values for the variable.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single trace

This will create a simple time series plot of BaG_A11 for all cut sizes.

```
da.plot.timeseries --y bnd A11a 2015-05-01 2015-05-03
```

da.allan

A plot of standard deviation versus averaging time.

Usage

da.allan [switches...] [[station] [variables] times [archive]|[file]]

Switches

--fit[=BOOLEAN]

This controls if the fit line is enabled.

Default: **Enabled**

--fit-end=NUMBER

The end bound of the data that contributes to the fit. Any data after this will be ignored in the fit.

Default: **100 seconds**

This option only accepts numbers strictly greater than 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--fit-start=NUMBER

The start bound of the data that contributes to the fit. Any data before this will be ignored in the fit.

Default: **10 seconds**

This option only accepts numbers strictly greater than 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top right corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--points[=BOOLEAN]

This enables the display of the discrete points.

Default: **Disabled**

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=STRING

This is the main graph title.

--x-title=STRING

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=VARIABLE

This is the variable or variables to the standard deviation of on the Y axis.

--y-title=STRING

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=VARIABLE

This is the variable or variables that determine the trace color. This generates a continuum of colors between minimum and maximum values for the variable.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single trace

This will create a simple Allan plot of PM10 data for BaG_A11.

```
da.show.allan --y sfa A11a 2012-12-03T02:13:44Z 2012-12-03T12:00:00Z
```

da.cdf

A cumulative distribution plot.

Usage

da.cdf [switches...] [[station] [variables] times [archive]|[file]]

Switches

--fit[=BOOLEAN]

This sets the display of the distribution fits.

Default: **Enabled**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft
The legend is located inside the trace area at the bottom left corner.

--legend=bottomright
The legend is located inside the trace area at the bottom right corner.

--legend=disabled
No legend is displayed.

--legend=inside
Same as --legend=**topleft**

--legend=left
The legend is located outside and to the left of the graph.

--legend=none
Same as --legend=**disabled**

--legend=off
Same as --legend=**disabled**

--legend=outside
Same as --legend=**right**

--legend=right
The legend is located outside and to the right of the graph.

--legend=topleft
The legend is located inside the trace area at the top left corner.

--legend=topright
The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN]
This changes the variable axes to be logarithmic.

--quantile=NUMBER
This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=**STRING**

This is the main graph title.

--x=**VARIABLE**

This is the variable or variables to plot on the X axis.

--x-title=**STRING**

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y-title=**STRING**

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=**VARIABLE**

This is the variable or variables that determine the trace color. This generates a continuum of colors between minimum and maximum values for the variable.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Simple

This will create a basic CDF plot.

```
da.show.cdf --log --x bnd A11a 2015-05-01 2015-05-03
```

da.cycle

A box-whisker plot of periodic cycles.

Usage

da.cycle [switches...] [[station] [variables] times [archive]|[file]]

Switches

--box[=BOOLEAN]

When disabled this hides the box and displays only the whiskers and median.

Default: **Enabled**

--division=INTERVALBLOCK

The X axis is divided based on this interval. For example, if this is set to one month then a division and bin appear for every month in the period. This cannot be greater than the total period. Alignment is ignored (implicitly aligned with the period).

Default: **One month**

When no units are specified the interval is assumed to be in months.

--extend[=BOOLEAN]

If set then the median lines of the bins are extended to cover the entire graph. If totals are enabled only the median of the totals is extended.

Default: **Enabled with a total bin**

--fit=TYPE

This is the type of fit displayed on the graph.

Default: **Disabled**

The possible values are:

--fit=2poly

Same as --fit=**secondorder**

--fit=3poly

Same as --fit=**thirdorder**

--fit=disabled

No fit is displayed.

--fit=leastsquares

Same as --fit=**linear**

--fit=linear

Linear least squares.

--fit=linearzero

Linear least squares forced through zero.

--fit=lsq

Same as --fit=**linear**

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--fit=secondorder

Second order polynomial.

--fit=secondorderzero

Second order polynomial forced through zero.

--fit=thirdorder

Third order polynomial.

--fit=thirdorderzero

Third order polynomial forced through zero.

--fit=zerolinear

Same as --fit=**linearzero**

--fit=zerosecondorder

Same as --fit=**secondorderzero**

--fit=zerothirdorder

Same as --fit=**thirdorderzero**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN**]**

This changes the variable axes to be logarithmic.

--period=INTERVALBLOCK

This is the total period of the display. Data is plotted such that the X axis repeats on this period.

Default: **One year**

When no units are specified the interval is assumed to be in years.

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=STRING

This is the main graph title.

--total[=BOOLEAN]

This enables the display of a bin that contains all data.

--x-title=STRING

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=VARIABLE

This is the variable or variables to plot on the Y axis.

--y-title=STRING

This is the title output along the Y axis.

Default: **Generated from the axis units.**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That

is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Hourly

This will create a daily cycle plot with bins for each hour of the day.

```
da.show.cycle --division --period --y bnd A11a 2015-05-01 2015-05-03
```

da.density

A two dimensional scatter plot of point density.

Usage

da.density [switches...] [[station] [variables] times [archive]|[file]]

Switches

--density-limit=NUMBER

The limit of the contribution of any point to the total density as a percentage of the total number

of points.

Default: **0.5% of data**

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--fit=TYPE

This is the type of fit displayed on the graph.

Default: **Disabled**

The possible values are:

--fit=2poly

Same as --fit=**secondorder**

--fit=3poly

Same as --fit=**thirdorder**

--fit=disabled

No fit is displayed.

--fit=leastsquares

Same as --fit=**linear**

--fit=linear

Linear least squares.

--fit=linearzero

Linear least squares forced through zero.

--fit=lsq

Same as --fit=**linear**

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--fit=secondorder

Second order polynomial.

--fit=secondorderzero

Second order polynomial forced through zero.

--fit=thirdorder

Third order polynomial.

--fit=thirdorderzero

Third order polynomial forced through zero.

--fit=zerolinear

Same as --fit=**linearzero**

--fit=zerosecondorder

Same as --fit=**secondorderzero**

--fit=zerothirdorder

Same as --fit=**thirdorderzero**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN**]**

This changes the variable axes to be logarithmic.

--points[=BOOLEAN**]**

This enables the display of the discrete points.

Default: **Disabled**

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=STRING

This is the main graph title.

--x=VARIABLE

This is the variable or variables to plot on the X axis.

--x-title=STRING

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=VARIABLE

This is the variable or variables to plot on the Y axis.

--y-title=STRING

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=VARIABLE

This is the variable or variables that determine the trace color. This generates a continuum of colors between minimum and maximum values for the variable. For example, it could be used to display the humidity for points comparing the scattering between two nephelometers.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut

sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single trace

This will create a simple density plot of BaG_A12 vs BaG_A11 for PM10 data.

```
da.show.density --x --y bnd BaG_A11,BaG_A12 2015-05-01 2015-05-03
```

da.layout

An already defined layout of other plots.

Usage

da.layout [switches...] [[station] [variables] times [archive]|[file]]

Switches

--configuration=SELECTION

The variable in the archive used to load the layout.

--input=**FILE**

If set then the layout is loaded from this file.

--path=**STRING**

The path within the configuration to use.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

Examples

Static file

This will load the layout contained in the file "layout.xml".

```
da.show.layout --input=layout.xml bnd S11a 2015-05-01 2015-05-03
```

Archive display

This a display from the main system configuration.

```
da.show.layout --path=Templates/Graphs/Scattering bnd S11a 2015-05-01 2015-05-03
```

da.pdf

A probability distribution plot.

Usage

da.pdf [switches...] [[station] [variables] times [archive]|[file]]

Switches

--bins=INTEGER

This is the number of bins spread uniformly across the data range used in the histogram.

Default: **10**

This option only accepts integers greater than or equal to 2 and less than or equal to 10000.

--fit=TYPE

This is the type of fit used on the histogram.

Default: **Cublic spline**

The possible values are:

--fit=cspline

Cubic spline interpolation.

--fit=cubicspline

Same as --fit=**cspline**

--fit=disabled

No fit is displayed.

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft
: The legend is located inside the trace area at the bottom left corner.

--legend=bottomright
: The legend is located inside the trace area at the bottom right corner.

--legend=disabled
: No legend is displayed.

--legend=inside
: Same as --legend=**topleft**

--legend=left
: The legend is located outside and to the left of the graph.

--legend=none
: Same as --legend=**disabled**

--legend=off
: Same as --legend=**disabled**

--legend=outside
: Same as --legend=**right**

--legend=right
: The legend is located outside and to the right of the graph.

--legend=topleft
: The legend is located inside the trace area at the top left corner.

--legend=topright
: The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN]
: This changes the variable axes to be logarithmic.

--quantile=NUMBER
: This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

  This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--show-bins[=BOOLEAN]
: This enables the display of the histogram bins used in the PDF.

--title=**STRING**

This is the main graph title.

--x=**VARIABLE**

This is the variable or variables to plot on the X axis.

--x-title=**STRING**

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y-title=**STRING**

This is the title output along the Y axis.

Default: **Generated from the axis units.**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Simple

This will create a basic PDF plot.

```
da.show.pdf --log --x bnd A11a 2015-05-01 2015-05-03
```

da.scatter

A two dimensional scatter plot of values.

Usage

da.scatter [switches...] [[station] [variables] times [archive]|[file]]

Switches

--fit=TYPE

This is the type of fit displayed on the graph.

Default: **Disabled**

The possible values are:

--fit=2poly

Same as --fit=**secondorder**

--fit=3poly

Same as --fit=**thirdorder**

--fit=disabled

No fit is displayed.

--fit=leastsquares

Same as --fit=**linear**

--fit=linear

Linear least squares.

--fit=linearzero

Linear least squares forced through zero.

--fit=lsq

Same as --fit=**linear**

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--fit=secondorder

Second order polynomial.

--fit=secondorderzero

Second order polynomial forced through zero.

--fit=thirdorder

Third order polynomial.

--fit=thirdorderzero

Third order polynomial forced through zero.

--fit=zerolinear

Same as --fit=**linearzero**

--fit=zerosecondorder

Same as --fit=**secondorderzero**

--fit=zerothirdorder

Same as --fit=**thirdorderzero**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN]

This changes the variable axes to be logarithmic.

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=STRING

This is the main graph title.

--x=VARIABLE

This is the variable or variables to plot on the X axis.

--x-title=STRING

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=VARIABLE

This is the variable or variables to plot on the Y axis.

--y-title=STRING

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=VARIABLE

This is the variable or variables that determine the trace color. This generates a continuum of

colors between minimum and maximum values for the variable. For example, it could be used to display the humidity for points comparing the scattering between two nephelometers.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single trace

This will create a simple scatter plot of BaG_A12 vs BaG_A11 for all cut sizes.

```
da.show.scatter --x --y bnd BaG_A11,BaG_A12 2015-05-01 2015-05-03
```

da.timeseries

A two dimensional time series of values.

Usage

da.timeseries [switches...] [[station] [variables] times [archive]|[file]]

Switches

--fit=TYPE

This is the type of fit displayed on the graph.

Default: **Disabled**

The possible values are:

--fit=2poly

Same as --fit=**secondorder**

--fit=3poly

Same as --fit=**thirdorder**

--fit=disabled

No fit is displayed.

--fit=leastsquares

Same as --fit=**linear**

--fit=linear

Linear least squares.

--fit=linearzero

Linear least squares forced through zero.

--fit=lsq

Same as --fit=**linear**

--fit=none

Same as --fit=**disabled**

--fit=off

Same as --fit=**disabled**

--fit=secondorder

Second order polynomial.

--fit=secondorderzero

Second order polynomial forced through zero.

--fit=thirdorder

Third order polynomial.

--fit=thirdorderzero

Third order polynomial forced through zero.

--fit=zerolinear

Same as --fit=**linearzero**

--fit=zerosecondorder

Same as --fit=**secondorderzero**

--fit=zerothirdorder

Same as --fit=**thirdorderzero**

--grid=TYPE

This is the grid mode to display. This enables or disables the possible grids that are shown.

Default: **Disabled**

The possible values are:

--grid=both

Both X and Y grid lines are displayed.

--grid=disabled

No grid is displayed.

--grid=enable

Same as --grid=**both**

--grid=enabled

Same as --grid=**both**

--grid=none

Same as --grid=**disabled**

--grid=off

Same as --grid=**disabled**

--grid=x

Only grid lines extending from the X axis are displayed.

--grid=x-y

Same as --grid=**both**

--grid=xy

Same as --grid=**both**

--grid=y

Only grid lines extending from the Y axis are displayed.

--legend=TYPE

This is the legend mode to use. This sets the position of the legend for the graph.

Default: **Inside top left corner**

The possible values are:

--legend=bottomleft

The legend is located inside the trace area at the bottom left corner.

--legend=bottomright

The legend is located inside the trace area at the bottom right corner.

--legend=disabled

No legend is displayed.

--legend=inside

Same as --legend=**topleft**

--legend=left

The legend is located outside and to the left of the graph.

--legend=none

Same as --legend=**disabled**

--legend=off

Same as --legend=**disabled**

--legend=outside

Same as --legend=**right**

--legend=right

The legend is located outside and to the right of the graph.

--legend=topleft

The legend is located inside the trace area at the top left corner.

--legend=topright

The legend is located inside the trace area at the top right corner.

--log[=BOOLEAN]

This changes the variable axes to be logarithmic.

--quantile=NUMBER

This causes the plot to only include the inner percentage of data set. For example, setting this to 99, causes the graph to exclude the top and bottom 0.5% of data.

This option only accepts numbers strictly greater than 0 and less than or equal to 100. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--title=STRING

This is the main graph title.

--x-title=STRING

This is the title output along the X axis.

Default: **Generated from the axis units.**

--y=VARIABLE

This is the variable or variables to plot on the Y axis.

--y-title=STRING

This is the title output along the Y axis.

Default: **Generated from the axis units.**

--z=VARIABLE

This is the variable or variables that determine the trace color. This generates a continuum of colors between minimum and maximum values for the variable.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are

explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Single trace

This will create a simple time series plot of BaG_A11 for all cut sizes.

```
da.show.timeseries --y bnd A11a 2015-05-01 2015-05-03
```

External Data Conversion

These programs are used to convert data from an external format to native CPD3 data. Most operate by taking a file as input and output converted data.

da.acquire.2b.ozone205

This component provides a parser and control to acquire data from a 2B Ozone Monitor model 205 instrument. This includes both online control and offline parsing.

Usage

da.acquire.2b.ozone205 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=**STRING**

The station code to assign to data values.

Default: **NIL**

--suffix=**STRING**

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=**INTEGER**

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=**INTEGER**

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.2b.ozone205 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.2b.ozone205 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data.

```
da.acquire.2b.ozone205
```

da.acquire.ad.cpcmagic200

This component provides a parser and control to acquire Aerosol Dynamics Magic 200 CPC data.

Usage

da.acquire.ad.cpcmagic200 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.ad.cpcmagic200 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.ad.cpcmagic200 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Ingest completely raw data

```
da.acquire.ad.cpcmagic200 --mode=raw
```

Convert data

```
da.acquire.ad.cpcmagic200
```

da.acquire.aerodyne.caps

This component provides a parser and control to acquire data from a Aerodyne CAPS instrument. This includes both online control and offline parsing.

Usage

da.acquire.aerodyne.caps [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--basetime=TIME

When dealing with files without a full timestamp on each record, this time is used to reconstruct the full time from the time of day.

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--wavelength=NUMBER

This is the wavelength in nm of the laser in the CAPS.

Default: **532nm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.aerodyne.caps --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.aerodyne.caps --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Ingest completely raw data

```
da.acquire.aerodyne.caps --mode=raw
```

Convert data.

```
da.acquire.aerodyne.caps
```

da.acquire.bmi.cpc1710

This component provides a parser and control to acquire BMI mixing CPC #1710 data. This includes both offline parsing and online control.

Usage

da.acquire.bmi.cpc1710 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--recalculate[=BOOLEAN]

When enabled the concentration is calculated from the count rate and reported flow.

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.bmi.cpc1710 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.bmi.cpc1710 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data using the instrument reported concentration

```
da.acquire.bmi.cpc1710
```

Recalculate the concentration from the count rate

```
da.acquire.bmi.cpc1710 --recalculate
```

da.acquire.bmi.cpc1720

This component provides a parser and control to acquire BMI mixing CPC #1720 data. This includes both offline parsing and online control.

Usage

da.acquire.bmi.cpc1720 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--recalculate[=BOOLEAN]

When enabled the concentration is calculated from the count rate and reported flow.

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.bmi.cpc1720 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.bmi.cpc1720 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data using the instrument reported concentration

```
da.acquire.bmi.cpc1720
```

Recalculate the concentration from the count rate

```
da.acquire.bmi.cpc1720 --recalculate
```

da.acquire.campbell.cr1000gmd

This component provides a parser and control to acquire data from a Campbell Scientific CR1000 with the GMD acquisition firmware. This includes both offline parsing and online control.

Usage

da.acquire.campbell.cr1000gmd [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=**INTEGER**

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=**STRING**

The station code to assign to data values.

Default: **NIL**

--suffix=**STRING**

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=**INTEGER**

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=**INTEGER**

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.campbell.cr1000gmd --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.campbell.cr1000gmd --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data with the default settings.

```
da.acquire.campbell.cr1000gmd
```

da.acquire.csd.pops

This component provides a parser and control to acquire data from a CSD POPS instrument. This includes both online control and offline parsing.

Usage

da.acquire.csd.pops [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=**STRING**

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=**INTEGER**

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=**INTEGER**

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--diameter=**NUMBER...**

This sets the center diameter of the bins in µm. When not set no information about bin sizes is generated.

This option requires at least one number if it is used.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.csd.pops --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.csd.pops --mode=twotime --record-index=4 --time-index=2 --time-index2=3
```

Convert data.

```
da.acquire.csd.pops
```

da.acquire.dmt.bcp

This component provides a parser and control to acquire DMT Backscatter Cloud Probe data. This includes both offline parsing and online control.

Usage

da.acquire.dmt.bcp [switches...] [file]

Switches

--mode=**TYPE**

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=**INTEGER**

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=**STRING**

The station code to assign to data values.

Default: **NIL**

--suffix=**STRING**

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--q=NUMBER

This is the flow rate of the CPC at lpm measured at ambient conditions.

Default: **1.4210 lpm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.dmt.bcp --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.dmt.bcp --mode=twotime --record-index=4 --time-index=2 --time-index2=3
```

Convert data using the default flow rate

```
da.acquire.dmt.bcp
```

Explicitly set flow rate.

```
da.acquire.dmt.bcp --q=25.78
```

da.acquire.dmt.ccn

This component provides a parser and control to acquire data from a DMT CCN. This includes both offline parsing and online control.

Usage

da.acquire.dmt.ccn [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--basetime=TIME

When dealing with files without a full timestamp on each record, this time is used to reconstruct the full time from the time of day on the H records.

--model-interval=INTERVAL

This is the interval the model is executed at. This defines the amount of data averaged together to make a single request for the model to generate a supersaturation for.

Default: **One minute**

The interval must be greater than zero in length.

--processing-interval=**INTERVAL**

This is the interval the that general processing is executed at. This defines the amount of time that enters into the reported dT standard deviation.

Default: **One minute**

The interval must be greater than zero in length.

--record-index=**INTEGER**

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--stability=**SMOOTHER**

This is the smoother that is used to by the acquisition system to determine if T3-T1 is stable.

Default: **Always stable**

The smoother is used for stability and spike detection.

--station=**STRING**

The station code to assign to data values.

Default: **NIL**

--suffix=**STRING**

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=**INTEGER**

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=**INTEGER**

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single

dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.dmt.ccn --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.dmt.ccn --mode=twotime --record-index=4 --time-index=2 --time-index2=3
```

Convert data

```
da.acquire.dmt.ccn
```

da.acquire.dmt.pax

This component provides a parser and control to acquire data from a DMT PAX instrument. This includes both online control and offline parsing.

Usage

da.acquire.dmt.pax [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--wavelength=NUMBER

This wavelength in nm of the laser in the PAX.

Default: **532nm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.dmt.pax --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.dmt.pax --mode=twotime --record-index=4 --time-index=2 --time-index2=3
```

Convert data.

```
da.acquire.dmt.pax
```

da.acquire.eigenbrodt.nmo191

This component provides a parser and control to acquire data from an Eigenbrodt NMO191 rain sampler. This includes both offline parsing and online control.

Usage

da.acquire.eigenbrodt.nmo191 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.eigenbrodt.nmo191 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.eigenbrodt.nmo191 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Ingest completely raw data

```
da.acquire.eigenbrodt.nmo191 --mode=raw
```

Convert data from the instrument

```
da.acquire.eigenbrodt.nmo191
```

da.acquire.ecotech.nephaurora

This component provides a parser and control to acquire Ecotech Aurora 3000 or 4000 series nephelometer data.

Usage

da.acquire.ecotech.nephaurora [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.ecotech.nephaurora --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.ecotech.nephaurora --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Ingest completely raw data

```
da.acquire.ecotech.nephaurora --mode=raw
```

Convert data

```
da.acquire.ecotech.nephaurora
```

da.acquire.generic.metar

This component provides a parser and control to acquire data from an ICAO METAR data source. This includes both online acquisition and offline parsing.

Usage

da.acquire.generic.metar [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--basetime=TIME

When dealing with files without a full timestamp on each record, this time is used to reconstruct the full time from day of month and time of day.

--elevation=NUMBER

This is the station elevation in meters used to convert altimeter settings to absolute pressures.

Default: **0m**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.generic.metar --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.generic.metar --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Ingest completely raw data

```
da.acquire.generic.metar --mode=raw
```

Convert data.

```
da.acquire.generic.metar
```

da.acquire.gmd.clap3w

This component provides a parser and control to acquire GMD three wavelength CLAP data. This includes both offline parsing and online control.

Usage

da.acquire.gmd.clap3w [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--filter-baseline=**SMOOTHER**

This defines the smoothing used to establish a baseline at the start of a filter.

Default: **90 second duration with <0.001 RSD**

The smoother is used for stability detection.

--record-index=**INTEGER**

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--spot-normalize=**SMOOTHER**

This defines the smoothing used to establish a normalization at the start of a spot.

Default: **60 seconds max, 30 seconds min, 8 second delay, <0.001 RSD**

The smoother is used for stability detection.

--station=**STRING**

The station code to assign to data values.

Default: **NIL**

--suffix=**STRING**

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=**INTEGER**

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=**INTEGER**

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--cal-q=**NUMBER**

This is the constant multiplier to the reported flow and volume. This is applied after the hardware calibrates the voltage to a flow rate and integrates that to a volume.

Default: **1.0**

This option only accepts numbers strictly greater than 0.

--spot1=**NUMBER**

This is the area for spot %1. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **19.9 mm²**

This option only accepts numbers strictly greater than 0.

--spot2=**NUMBER**

This is the area for spot %1. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **19.9 mm²**

This option only accepts numbers strictly greater than 0.

--spot3=**NUMBER**

This is the area for spot %1. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **19.9 mm²**

This option only accepts numbers strictly greater than 0.

--spot4=**NUMBER**

This is the area for spot %1. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **19.9 mm²**

This option only accepts numbers strictly greater than 0.

--spot5=**NUMBER**

This is the area for spot %1. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **19.9 mm²**

This option only accepts numbers strictly greater than 0.

--spot6=NUMBER

This is the area for spot %1. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **19.9 mm²**

This option only accepts numbers strictly greater than 0.

--spot7=NUMBER

This is the area for spot %1. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **19.9 mm²**

This option only accepts numbers strictly greater than 0.

--spot8=NUMBER

This is the area for spot %1. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **19.9 mm²**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.gmd.clap3w --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.gmd.clap3w --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data with the default areas and calibrations

```
da.acquire.gmd.clap3w
```

Explicitly defined areas with a flow calibration

```
da.acquire.gmd.clap3w --cal-q=1.004 --spot1=19.74 --spot2=19.71 --spot3=20.05
--spot4=19.25 --spot5=19.76 --spot6=19.65 --spot7=19.47 --spot8=19.88
```

da.acquire.gmd.cpcpulse

This component provides a parser and control to acquire data from a GMD pulse counter box. This is normally paired with a TSI 3760 CPC to generate actual number concentrations. This includes both offline parsing and online control.

Usage

da.acquire.gmd.cpcpulse [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--channel=INTEGER

This is the input channel number on the acquisition box.

Default: **The first channel**

This option only accepts integers greater than or equal to 1 and less than or equal to 2.

--measuredtime[=BOOLEAN]

If set then the measured time interval between reports is used instead of assuming one second. This is generally not recommended because it is affected by processing delays and jitter in the system scheduling.

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--q=NUMBER

This is the flow rate of the CPC at lpm measured at ambient conditions.

Default: **1.4210 lpm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.gmd.cpcpulse --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.gmd.cpcpulse --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data from the first channel with the default flow rate.

```
da.acquire.gmd.cpcpulse
```

Explicitly set flow rate acquired from the second input channel.

```
da.acquire.gmd.cpcpulse --channel=2 --q=25.78
```

da.acquire.grimm.opc110x

This component provides a parser and control to acquire Grimm Dust Monitor 1.104 - 1.109 data. This includes both offline parsing and online control.

Usage

da.acquire.grimm.opc110x [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.grimm.opc110x --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.grimm.opc110x --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data

```
da.acquire.grimm.opc110x
```

da.acquire.magee.aethalometer162131

This component provides a parser and control to acquire data from a Magee Aethalometer, models AE16, AE21, or AE31. This includes both offline parsing and online control.

Usage

da.acquire.magee.aethalometer162131 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--bcunits=TYPE

This specifies the units that the instrument is set to report the EBC in.

Default: **Autodetect from the data**

The possible values are:

--bcunits=ng

ng/m³

--bcunits=ug

µg/m³

--p=NUMBER

This is the sample pressure the aethalometer is set to. The value is used to convert to an STP of zero °C and 1013.25 hPa.

Default: **1013 hPa**

This option only accepts numbers strictly greater than 10.

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--t=NUMBER

This is the sample temperature the aethalometer is set to. The value is used to convert to an STP of zero °C and 1013.25 hPa.

Default: **20 °C**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--timebase=INTEGER

This is the number of minutes expected between records.

Default: **5 minutes**

This option only accepts integers greater than or equal to 1.

--spot=NUMBER

This is the sampling area of the filter spot. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **50.0 mm²**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.magee.aethalometer162131 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.magee.aethalometer162131 --mode=twotime --record-index=4 --time-index=2
--time-index2=3
```

Ingest completely raw data

```
da.acquire.magee.aethalometer162131 --mode=raw
```

Convert data using the default high sensitivity spot size.

```
da.acquire.magee.aethalometer162131
```

Explicitly set the sample spot size.

```
da.acquire.magee.aethalometer162131 --spot=42.12
```

da.acquire.magee.aethalometer33

This component provides a parser and control to acquire data from a Magee Aethalometer models AE33. This includes both offline parsing and online control.

Usage

da.acquire.magee.aethalometer33 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--spot1=NUMBER

This is the sampling area of the first spot. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **78.5 mm²**

This option only accepts numbers strictly greater than 0.

--spot2=NUMBER

This is the sampling area of the second spot. If it is greater than 0.1 it is assumed to be in mm², otherwise it is treated as being in m².

Default: **78.5 mm²**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.magee.aethalometer33 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.magee.aethalometer33 --mode=twotime --record-index=4 --time-index=2
--time-index2=3
```

Ingest completely raw data

```
da.acquire.magee.aethalometer33 --mode=raw
```

Convert data using the default spot sizes.

```
da.acquire.magee.aethalometer33
```

Explicitly set the sample spot size.

```
da.acquire.magee.aethalometer33 --spot1=42.12
```

da.acquire.magee.tca08

This component provides a parser and control to acquire Magee TCA 08 data.

Usage

da.acquire.magee.tca08 [switches...] [file]

Switches

--mode=**TYPE**

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1
: CPD1 raw data recorder output

--mode=cpd2
: CPD2 raw data recorder output

--mode=doy
: Same as --mode=**twotime**

--mode=raw
: Raw data input, with implicit times

--mode=singletime
: CSV input with a single time field

--mode=twotime
: CSV input with a two part time field

--mode=yeardoy
: Same as --mode=**twotime**

--record-index=INTEGER
: The index, starting from one, of the record data.

  This option only accepts integers greater than or equal to 1.

--station=STRING
: The station code to assign to data values.

  Default: **NIL**

--suffix=STRING
: The instrument suffix (e.x. "S11") to use when completing variables.

  Default: **Automatically determined**

--time-index=INTEGER
: The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

  This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER
: The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.magee.tca08 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.magee.tca08 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Ingest completely raw data

```
da.acquire.magee.tca08 --mode=raw
```

Convert data

```
da.acquire.magee.tca08
```

da.acquire.maycomm.tdl

This component provides a parser to acquire data from a MayComm TDL. This includes both online control and offline parsing.

Usage

da.acquire.maycomm.tdl [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.maycomm.tdl --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.maycomm.tdl --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data

```
da.acquire.maycomm.tdl
```

da.acquire.pms.lasair

This component provides a parser and control to acquire data from a PMS LASAIR OPC instrument. This includes both online control and offline parsing.

Usage

da.acquire.pms.lasair [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--wavelength=NUMBER

This wavelength in nm of the laser in the LASAIR.

Default: **532nm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.pms.lasair --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.pms.lasair --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data.

```
da.acquire.pms.lasair
```

da.acquire.purpleair.pa2

This component provides a parser to interpret JSON data from a PurpleAir PA-II sensor. JSON data are usually provided in response to HTTP queries.

Usage

da.acquire.purpleair.pa2 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.purpleair.pa2 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.purpleair.pa2 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Ingest completely raw data

```
da.acquire.purpleair.pa2 --mode=raw
```

Convert data

```
da.acquire.purpleair.pa2
```

da.acquire.rmy.wind86xxx

This component provides a parser and control to acquire data from a R.M. Young 86xxx series sonic anemometers. This includes both online control and offline parsing.

Usage

da.acquire.rmy.wind86xxx [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.rmy.wind86xxx --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.rmy.wind86xxx --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data.

```
da.acquire.rmy.wind86xxx
```

da.acquire.rr.neph903

This component provides a parser and control to acquire data from a Radiance Research model M903 nephelometer. This includes both online control and offline parsing.

Usage

da.acquire.rr.neph903 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--wavelength=NUMBER

This wavelength in nm of the optical filter in the nephelometer.

Default: **530nm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single

dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.rr.neph903 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.rr.neph903 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data.

```
da.acquire.rr.neph903
```

da.acquire.rr.psap1w

This component provides a parser and control to acquire Radiance Research single wavelength PSAP. This includes both offline parsing and online control.

Usage

da.acquire.rr.psap1w [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--cal-q=COEFFICIENT...

This is the calibration applied to the flow reported by the instrument.

Constant values are accepted.

--disable-recovery[=BOOLEAN]

When set, this disables recovery of the normalization values from the instrument reported transmittance. This can be useful if the reported transmittance is not valid.

--filter-change=SMOOTHER

This defines the smoothing used to detect the start and end of filter changes.

Default: **60 seconds max, 30 seconds min, <0.02 RSD, 2.0 spike factor**

The smoother is used for stability and spike detection.

--filter-normalize=SMOOTHER

This defines the smoothing used to establish a normalization at the start of the filter.

Default: **60 seconds max, 30 seconds min, 8 second delay, <0.001 RSD**

The smoother is used for stability detection.

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--spot=NUMBER

This is the area of the sample spot. If it is greater than 0.1 it is assumed to be in mm^2, otherwise it is treated as being in m^2.

Default: **17.83 mm^2**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.rr.psap1w --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.rr.psap1w --mode=twotime --record-index=4 --time-index=2 --time-index2=3
```

Ingest completely raw data

```
da.acquire.rr.psap1w --mode=raw
```

Convert data with the default area and calibrations

```
da.acquire.rr.psap1w
```

Explicitly defined area

```
da.acquire.rr.psap1w --spot=18.01
```

da.acquire.rr.psap3w

This component provides a parser and control to acquire Radiance Research three wavelength PSAP. This includes both offline parsing and online control.

Usage

da.acquire.rr.psap3w [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--cal-q=COEFFICIENT...

This is the calibration applied to the flow reported by the instrument.

Constant values are accepted.

--disable-recovery[=BOOLEAN]

When set, this disables recovery of the normalization values from the instrument reported transmittance. This can be useful if the reported transmittance is not valid.

--filter-change=SMOOTHER

This defines the smoothing used to detect the start and end of filter changes.

Default: **60 seconds max, 30 seconds min, <0.02 RSD, 2.0 spike factor**

The smoother is used for stability and spike detection.

--filter-normalize=SMOOTHER

This defines the smoothing used to establish a normalization at the start of the filter.

Default: **60 seconds max, 30 seconds min, 8 second delay, <0.001 RSD**

The smoother is used for stability detection.

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--spot=NUMBER

This is the area of the sample spot. If it is greater than 0.1 it is assumed to be in mm^2, otherwise it is treated as being in m^2.

Default: **17.83 mm²**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.rr.psap3w --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.rr.psap3w --mode=twotime --record-index=4 --time-index=2 --time-index2=3
```

Ingest completely raw data

```
da.acquire.rr.psap3w --mode=raw
```

Convert data with the default area and calibrations

```
da.acquire.rr.psap3w
```

Explicitly defined area

```
da.acquire.rr.psap3w --spot=18.01
```

da.acquire.thermo.maap5012

This component provides a parser and control to acquire data from a Thermo MAAP model 5012. This includes both offline parsing and online control.

Usage

da.acquire.thermo.maap5012 [switches...] [file]

Switches

--mode=**TYPE**

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--cal-q=**COEFFICIENT...**

This is the calibration applied to the flow reported by the instrument.

Constant values are accepted.

--filter-change=**SMOOTHER**

This defines the smoothing used to detect the start of filter changes. When not set only the flags are used in detection.

The smoother is used for stability and spike detection.

--record-index=**INTEGER**

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=**STRING**

The station code to assign to data values.

Default: **NIL**

--suffix=**STRING**

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=**INTEGER**

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=**INTEGER**

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--spot=**NUMBER**

This is the area of the sample spot. If it is greater than 0.1 it is assumed to be in mm^2, otherwise it is treated as being in m^2.

Default: **200 mm^2**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.thermo.maap5012 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.thermo.maap5012 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Ingest completely raw data

```
da.acquire.thermo.maap5012 --mode=raw
```

Convert data

```
da.acquire.thermo.maap5012
```

da.acquire.tsi.cpc3010

This component provides a parser and control to acquire TSI model 3010 CPC data. This includes both offline parsing and online control.

Usage

da.acquire.tsi.cpc3010 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--measuredtime[=BOOLEAN**]**

If set then the measured time interval between reports is used instead of assuming one second.

This is generally not recommended because it is affected by processing delays and jitter in the system scheduling.

--record-index=**INTEGER**

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=**STRING**

The station code to assign to data values.

Default: **NIL**

--suffix=**STRING**

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=**INTEGER**

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=**INTEGER**

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--q=**NUMBER**

This is the flow rate of the CPC at lpm measured at ambient conditions.

Default: **1.0 lpm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.tsi.cpc3010 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.tsi.cpc3010 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data with the default flow rate.

```
da.acquire.tsi.cpc3010
```

Explicitly set flow rate.

```
da.acquire.tsi.cpc3010 --q=1.05
```

da.acquire.tsi.cpc302x

This component provides a parser and control to acquire TSI model 3022A and 3025A CPCs data. This includes both offline parsing and online control.

Usage

da.acquire.tsi.cpc302x [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--measuredtime[=BOOLEAN]

If set then the measured time interval between reports is used instead of assuming one second. This is generally not recommended because it is affected by processing delays and jitter in the system scheduling.

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--q=NUMBER

This is the sample flow rate of the CPC at lpm measured at ambient conditions.

Default: **0.03 lpm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.tsi.cpc302x --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.tsi.cpc302x --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data with the default flow rate.

```
da.acquire.tsi.cpc302x
```

Explicitly set flow rate.

```
da.acquire.tsi.cpc302x --q=0.032
```

da.acquire.tsi.cpc377x

This component provides a parser and control to acquire TSI model 3771-3776, and 3790 CPCs. This includes both offline parsing and online control.

Usage

da.acquire.tsi.cpc377x [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--q=NUMBER

This is the sample flow rate of the CPC at lpm measured at ambient conditions.

Default: **0.03 lpm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.tsi.cpc377x --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.tsi.cpc377x --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data with the default flow rate.

```
da.acquire.tsi.cpc377x
```

Explicitly set flow rate.

```
da.acquire.tsi.cpc377x --q=0.3
```

da.acquire.tsi.cpc3781

This component provides a parser and control to acquire data from a TSI 3781 water based CPC. This includes both offline parsing and online control.

Usage

da.acquire.tsi.cpc3781 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--q=NUMBER

This is the sample flow rate of the CPC at lpm measured at ambient conditions.

Default: **0.12 lpm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.tsi.cpc3781 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.tsi.cpc3781 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data

```
da.acquire.tsi.cpc3781
```

da.acquire.tsi.cpc3783

This component provides a parser and control to acquire data from a TSI 3783 water based CPC. This includes both offline parsing and online control.

Usage

da.acquire.tsi.cpc3783 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--q=NUMBER

This is the sample flow rate of the CPC at lpm measured at ambient conditions.

Default: **0.12 lpm**

This option only accepts numbers strictly greater than 0.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.tsi.cpc3783 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.tsi.cpc3783 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data

```
da.acquire.tsi.cpc3783
```

da.acquire.tsi.mfm4xxx

This component provides a parser and control to acquire data from TSI Mass Flow Meters models 4000/4100. This includes both offline parsing and online control.

Usage

da.acquire.tsi.mfm4xxx [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--binary-flow=NUMBER

This is the constant scalar used to convert binary flow measurements to lpm units. For 4000 series meters this should be 100 and for 4100 series meters it should be 1000.

Default: **100.0**

This option only accepts numbers strictly greater than 0.

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.tsi.mfm4xxx --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.tsi.mfm4xxx --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data from the instrument

```
da.acquire.tsi.mfm4xxx
```

da.acquire.tsi.neph3563

This component provides a parser and control to acquire data from a TSI 3563 Integrating Nephelometer. This includes both offline parsing and online control.

Usage

da.acquire.tsi.neph3563 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Raw data**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=raw

Raw data input, with implicit times

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.tsi.neph3563 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.tsi.neph3563 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data

```
da.acquire.tsi.neph3563
```

da.acquire.vaisala.pwdx2

This component provides a parser and control to acquire data from a Vaisala PWD12, PWD22, and PWD52 instruments. This includes both online control and offline parsing.

Usage

da.acquire.vaisala.pwdx2 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.vaisala.pwdx2 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.vaisala.pwdx2 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data.

```
da.acquire.vaisala.pwdx2
```

da.acquire.vaisala.wmt700

This component provides a parser and control to acquire data from a Vaisala WMT700 sonic anemometers. This includes both online control and offline parsing.

Usage

da.acquire.vaisala.wmt700 [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=STRING

The station code to assign to data values.

Default: **NIL**

--suffix=STRING

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=INTEGER

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=INTEGER

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.vaisala.wmt700 --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.vaisala.wmt700 --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data.

```
da.acquire.vaisala.wmt700
```

da.acquire.vaisala.wxt5xx

This component provides a parser and control to acquire data from a Vaisala WXTxx series instrument. This includes both online control and offline parsing.

Usage

da.acquire.vaisala.wxt5xx [switches...] [file]

Switches

--mode=TYPE

This is the base import mode. It defines the defaults for other options. Thos defaults may be overridden by setting those options.

Default: **Single time field**

The possible values are:

--mode=cpd1

CPD1 raw data recorder output

--mode=cpd2

CPD2 raw data recorder output

--mode=doy

Same as --mode=**twotime**

--mode=singletime

CSV input with a single time field

--mode=twotime

CSV input with a two part time field

--mode=yeardoy

Same as --mode=**twotime**

--record-index=INTEGER

The index, starting from one, of the record data.

This option only accepts integers greater than or equal to 1.

--station=**STRING**

The station code to assign to data values.

Default: **NIL**

--suffix=**STRING**

The instrument suffix (e.x. "S11") to use when completing variables.

Default: **Automatically determined**

--time-index=**INTEGER**

The index, starting from one, of the first (and only unless using multiple fields) time field in each record.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

--time-index2=**INTEGER**

The index, starting from one, of the second time field.

This option only accepts integers greater than or equal to 1. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Ingest CPD1 raw data record output

```
da.acquire.vaisala.wxt5xx --mode=cpd1
```

Discard the first field and ingest year and DOY times

```
da.acquire.vaisala.wxt5xx --mode=twotime --record-index=4 --time-index=2 --time
-index2=3
```

Convert data.

```
da.acquire.vaisala.wxt5xx
```

da.generate.fromebas

This component generates data by downloading available data from the NILU/EBAS/WDCA archive. Because the download size can be large there can be a long waiting time for data to begin.

Usage

da.generate.fromebas [switches...] station... times

Switches

--profile=**STRING**

This is the profile name to generate data for. Different profiles can generate different sets of data. Consult the station configuration for available profiles.

Default: **aerosol**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

Examples

Generate fromebas data

This will generate fromebas data for the "aerosol" profile.

```
da.generate.fromebas
```

da.import.cpd2

This component provides a mechanism to directly ingest CPD2 style data.

Usage

da.import.cpd2 [switches...] [file]

Switches

--archive=**STRING**

This is the archive that the data is imported as. For example, "raw" or "clean"

Default: **raw**

--enable-cut-flags[=**BOOLEAN]**

If set then any available cut size flags are translated.

Default: **Enabled**

--translate-arrays[=**BOOLEAN]**

If set then variables that would form arrays in CPD3 are translated accordingly. For example, size distributions are converted to CPD3 arrays.

Default: **Enabled**

--translate-ends[=**BOOLEAN]**

If set then variables that have difference measurement ends are translated to CPD3 equivalents. For example, transmitance measurements.

Default: **Enabled**

--translate-legacy-flags[=**BOOLEAN]**

If set then translation is performed on legacy CPD1 style flags. This translates the bits into semantic flags for CPD3 processing.

Default: **Enabled**

--translate-neph-flags[=**BOOLEAN]**

If set then translation is performed CPD2 nephelometers flags. This translates the bits into semantic flags for CPD3 processing.

Default: **Enabled**

--translate-psap-flags[=**BOOLEAN]**

If set then translation is performed CPD2 PSAP or CLAP flags. This translates the bits into semantic flags for CPD3 processing.

Default: **Enabled**

--translate-system-flags[=**BOOLEAN]**

If set then translation is performed CPD2 style system flags. This translates the bits into semantic flags for CPD3 processing.

Default: **Enabled**

--buffer=INTEGER

This is the maximum number of instances of a type of record that can be buffered before output is forced. This limit only takes effect if a record type stops appearing, which forces all other records to be buffered.

Default: **10000**

This option only accepts integers greater than or equal to 2.

--wait=NUMBER

This is the maximum number of seconds to wait for data before flushing buffers. This is used to keep realtime data output responsive, but it can be disabled (set to undefined) for batch output. A shorter wait time increased the amount of overhead generated in the output.

Default: **1 second**

This option only accepts numbers greater than or equal to 0 and less than or equal to 86400. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Defaults

This imports CPD2 data using all flags translation and sets the archive to raw.

```
da.import.cpd2
```

da.import.ebas

This component provides a mechanism to directly ingest EBAS style NASA/Ames formatted data. It operates by matching patterns in the file against a set of pre-configured templates and generating the data they match.

Usage

da.import.ebas [switches...] [file]

Switches

--archive=STRING

This is the archive that the data is imported as. For example, "raw" or "clean". This can be overridden by specific import profiles.

Default: **raw**

--instrument=STRING

This is the instrument suffix to use when generating variable names. For example, "S11". This can be override by specific profiles.

Default: **Automatic**

--profile=STRING

This is the base profile configuration used to convert data. The profile determines the templates the data are matched against to produce the final output. Consult the station configuration for available profiles.

Default: **aerosol**

--station=STRING

This is the station that the data is imported as. For example, "bnd". This can be overridden by specific import profiles. Normally this is determined by the contents of the file.

Default: **Automatic**

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Defaults

This imports EBAS data using the normal profile and fully automatic determination.

```
da.import.ebas
```

Internal System Programs

These programs are normally only used for internal operations. They are rarely called directly by a user.

da.archive

This component provides an interface to write a stream of data to the main archive.

Usage

da.archive [switches...] [[station] variables times [archive]|[file]]

Switches

--quiet[=BOOLEAN]

If set then no progress output is displayed.

Default: **Disabled**

--remove=TYPE

This defines the data explicitly removed as part of the write. Identical (same station, archive, variable, flavors, start, end, and priority) values are always replaced and do not need removal. Data in the event archive is never removed.

Default: **Remove values at all priorities within the updated time range**

The possible values are:

--remove=instrument

Remove all values of an instrument (unique station, archive, and suffix) of any priority within the range the update of that instrument spans.

--remove=instrument-priority-zero

Remove all values of an instrument (unique station, archive, and suffix) with priority zero within the range the update of that instrument spans.

--remove=none

Do not explicitly remove any values, only directly replace any inputs.

--remove=purge

Remove all values seen in the input from the archive entirely.

--remove=purge-instrument

Remove all values of an instrument (unique station, archive, and suffix) of any priority within the range the of data while also removing any values seen in the input.

--remove=purge-variable

Remove all values of a variable (unique station, archive, and variable) of any priority within the range of data while also removing any values seen in the input.

--remove=variable

Remove all values of a variable (unique station, archive, and variable) of any priority within the range the update of that variable spans.

--remove=variable-priority-zero

Remove all values of a variable (unique station, archive, and variable) with priority zero within the range the update of that variable spans.

--remove-flavors=TYPE

This defines how flavors are considered during data removal.

Default: **Ignore flavors and only consider the other components**

The possible values are:

--remove-flavors=any

Ignore flavors and only consider the other components.

--remove-flavors=exact

Require an exact match of all flavors for removal.

--remove-flavors=flatten

Require a match of flavors but consider secondary components (e.x. coverage) as part of their main variable.

--run-tasks[=BOOLEAN]

If set then the immediate tasks are run after the archive add.

Default: **Enabled**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or

more variable specifications separated by , (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply BsB_S11 selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead ::BsB_S11:pm10 is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as bnd:BsB_S11,thd:BsB_S11 allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value everything can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as raw or clean_meta. The argument accepts multiple archives specified as regular expressions and separated by : or ; or ,. For example raw(_meta)? selects both the raw and raw_meta archives.

The special value allarchives may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

Examples

Defaults

This will add data to the archive, merging the metadata with any that already exists to form a continuous record. Data are replaced on a per-variable basis.

```
da.archive input.dat
```

da.email.send

This component sends a report of all new processing since the last time it was run. This report is delivered to a set of email addresses defined by the processing profile.

Usage

da.email.send [switches...] [station...] [times]

Switches

--profile=STRING

This is the profile name to execute. Multiple profiles can be defined to allow for different email types from a single station. Consult the station configuration for available profiles.

Default: **aerosol**

--quiet[=BOOLEAN]

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--wait=NUMBER

This is the maximum time to wait for another process to complete, in seconds. If there are two concurrent processes for the same station and profile, the new one will wait at most this many seconds before giving up without sending an email new data. If this is set to undefined, it will wait forever for the other process to complete.

Default: **One hour**

This option only accepts numbers greater than or equal to 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

Examples

Email the latest data report

This will generate and email out the latest processing report for the "aerosol" profile.

```
da.email.send 2015-05-01 2015-05-03
```

da.maintenance.archive

This component performs periodic maintenance on the data archive. This includes removing unused references and optimizing the database backend. This maintenance can take a very long time, so it is not recommended to do interactively.

Usage

da.maintenance.archive [switches...]

Switches

--cleanup-stale[=BOOLEAN]

When enabled, the maintenance will release all stale locks in the archive.

Default: **Enabled**

--optimize-database[=BOOLEAN]

When enabled, the maintenance will perform automated optimization and cleaning of the SQL database backend.

--quiet[=BOOLEAN]

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--recompress[=BOOLEAN]

When enabled, the maintenance will recompress any data that has not been modified for some time using a higher compression algorithm. This reduces the total size of the database and speeds up access, but can take a very long time.

--recompress-age=INTERVALBLOCK

This sets the minimum age of data to recompress. Data that has not been modified for this long will be recompressed.

Default: **Six months**

When no units are specified the interval is assumed to be in months.

--remove-unused[=BOOLEAN]

When enabled, the maintenance will remove unused references in in the archive.

--split-archive[=BOOLEAN]

This option causes the storage to be restructured so each archive is separated into a different table. Under some database backends, this allows for better concurrency at the small cost of

more complicated initialization.

--split-station[=BOOLEAN]

This option causes the storage to be restructured so each station is separated into a different table. Under some database backends, this allows for better concurrency at the small cost of more complicated initialization.

--split-variable[=BOOLEAN]

This option causes the storage to be restructured so each variable is separated into a different table. Under some database backends, this allows for better concurrency at the small cost of more complicated initialization.

Examples

Remove unused references

This will remove all unused references in the database backend.

```
da.maintenance.archive --remove-unused
```

da.output.sqldb

This writes data to an external SQL based database. The specific data written as well as the database target are controlled by the profile selected.

Usage

da.output.sqldb [switches...] station... times

Switches

--profile=STRING

This is the profile name to output. Multiple profiles can be defined to specify different sets of data that can be output independently. Consult the station configuration for available profiles.

Default: **aerosol**

--quiet[=BOOLEAN]

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular

expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

Examples

Output SQL data

This will output data for the default "aerosol" profile.

```
da.output.sqldb bnd 2015-05-01 2015-05-03
```

da.output.upload

This generates files of updated data and uploads them to external targets. Generation is performed by other components and all the results that they create are uploaded to their destinations.

Usage

da.output.upload [switches...] [station...] [times]

Switches

--after=TIME

This sets the time to re-upload after. When set, this will upload all data after the given time. This can be used to ignore the record of the last update and all prior uploaded ranges, but doing so can cause there to be updated data that does not exist in the final upload. No effect when used with an explicit time range.

Default: **Last run time**

Undefined values are permitted, to indicate that the option is set but not to a specific time.

--discard[=BOOLEAN**]**

When enabled this causes the generated files to be discarded instead of actually uploaded.

--profile=STRING

This is the profile name to perform. Multiple profiles can be defined to perform different types of uploading for a station. Consult the station configuration for available profiles.

Default: **aerosol**

--quiet[=BOOLEAN]

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--single[=BOOLEAN]

When enabled this results in only a single upload of all affected data, instead of the normal multiple segments used.

--wait=NUMBER

This is the maximum time to wait for another upload process to complete, in seconds. If there are two concurrent processes for the same station and profile, the new one will wait at most this many seconds before giving up without uploading new data. If this is set to undefined, it will wait forever for the other process to complete. This is only valid when no times are specified.

Default: **30 seconds**

This option only accepts numbers greater than or equal to 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

Examples

Upload data

This will upload data for the default "aerosol" profile.

```
da.output.upload 2015-05-01 2015-05-03
```

da.process.authorize

This component provides a simple interface to add authorization to process incoming data. It can operate by either adding the certificate digest directly or by scanning files and adding the

authorization needed to process them. When no stations or certificates (from files or otherwise) are given it inspects the current directory for data to authorize. If a station is given but no certificates, it looks for unauthorized data for that station.

Usage

da.process.authorize [switches...] [station...]

Switches

--certificate=**FILE**

This is the decryption certificate used with the data.

--deauthorize[=**BOOLEAN]**

When enabled any existing authorization is removed and replaced with the new authorization. This ensures that old data is not allowed to be accepted after a short grace period.

Default: **Enabled**

--digest=**STRING**

This specifies the digest of the certificate to add authorization for. When absent, this is calculated from the unauthorized files. Use "Unsigned" for data without an authorization certificate or "Implied" to set the authorization for already exchanged certificates.

--file=**FILE**

This will add the authorization needed to process the file.

--move[=**BOOLEAN]**

When enabled any files with that have their authorization added are automatically moved for reprocessing.

Default: **Enabled**

--profile=**STRING**

This is the profile name inspect. This is used when loading existing unauthorized files. Multiple profiles can be defined to perform different types of processing for a station. Consult the station processing configuration for available profiles.

Default: **aerosol**

--quiet[=**BOOLEAN]**

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--template=**STRING**

This specifies the name of the template configuration used to configure the authorization.

Default: **"Standard"**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

Examples

Authorize data in the current directory

This will change the authorization to allow for processing of all files in the current directory.

```
da.process.authorize
```

Change authorization

This will add authorization for any stations specified to only accept the given certificate digest.

```
da.process.authorize
--digest=57dfe3dfbb14c8e43a2f9e95ff5758686cb2f1d20a6afde77e2b577fd1a9fead972eaff5b4
38963a92ed7cd2a50fc044f6ddd02013e6766381783a03dd201891
```

File authorization

This will add authorization for the given file.

```
da.process.authorize --file=/tmp/sfa_authorized.cpd3
```

da.process.incoming

This component processes data and inserts it into the local archive. This usually takes the form of data transfer files uploaded by acquisition systems. It can also apply various forms of conversion of data files before the ingest. This differs from synchronization in that the incoming data are assumed be be newly created.

Usage

da.process.incoming [switches...] [station...]

Switches

--after=TIME

This sets the time to reprocess after. When set, this will process all files modified after the time. This also triggers a state reset when used. When set to undefined, all data are reprocessed.

Default: **Last run time**

Undefined values are permitted, to indicate that the option is set but not to a specific time.

--profile=**STRING**

This is the profile name to perform. Multiple profiles can be defined to perform different types of processing for a station. Consult the station configuration for available profiles.

Default: **aerosol**

--quiet[=**BOOLEAN]**

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--wait=**NUMBER**

This is the maximum time to wait for another upload process to complete, in seconds. If there are two concurrent processes for the same station and profile, the new one will wait at most this many seconds before giving up without uploading new data. If this is set to undefined, it will wait forever for the other process to complete.

Default: **30 seconds**

This option only accepts numbers greater than or equal to 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

Examples

Process all data

This will process all incoming data.

```
da.process.incoming
```

da.process.unpack

This component provides a mechanism to unpack the data contained in the internal files used to transfer data.

Usage

da.process.unpack [switches...] [file]

Switches

--certificate=**STRING**

This is the certificate used on the transfer file, if needed.

--deleted[=**BOOLEAN]**

When enabled, values are created for all erasure markers in the transfer file. This disables normal data output, so all the values produced are erasure markers.

--key=**STRING**

This is the key used on the transfer file, if needed.

--modified[=**BOOLEAN]**

When enabled, the values in the output are replaced with their modification times.

Arguments

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

If no file is supplied then the input is read from standard input.

Examples

Defaults

This unpacks the given transfer archive file.

```
da.process.unpack
```

da.smooth.realtime

This applies the realtime averager to input data. This is normally not used in an interactive context but can be part of a realtime processing chain. The output data will contain the three realtime archives (raw, rt_instant, and rt_boxcar). This will ignore the times specified on the data and re-assign them to the current time for realtime streaming. Note that some types of averaging require

inputs beyond the values being averaged. For example, to average raw absorptions, transmittance and path length are required because the average is performed with a difference measurement.

Usage

da.smooth.realtime [switches...] [[station] variables times [archive]|[file]]

Switches

--interval=**OFFSET**

This is width of the bin to average on. All data within each interval is averaged together to produce the output. An undefined interval averages all available data together.

Default: **One minute**

When no units are specified the offset is assumed to be in minutes. Unless explicitly disabled, the offset is assumed to be aligned. The interval must be greater than or equal to zero length.

--statistics[=**BOOLEAN]**

This enables statistics generation for all the averaged outputs.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

Examples

Default

This creates a realtime smoother that generates data on one minute interval.

```
da.smooth.realtime bnd S11a 2015-05-01 2015-05-03
```

Or:

```
da.smooth.realtime input_file.c3d
```

Or:

```
da.get bnd S11a 2015-05-01 2015-05-03 | da.smooth.realtime
```

da.sync.authorize

This component provides a simple interface to add authorizations to synchronize data. It can operate either by authorizing certificates from files or by directly specifying the certificate digests to use.

Usage

da.sync.authorize [switches...] station...

Switches

--certificate=FILE

This is the certificate used for synchronization.

--deauthorize[=BOOLEAN]

When enabled any existing authorization is removed and replaced with the new authorization. This ensures that the old synchronization certificate can no longer be used.

--digest=STRING

This specifies the digest of the certificate to add authorization for. When absent this is calculated from any certificates that are added.

--quiet[=BOOLEAN]

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--template=STRING

This specifies the name of the template configuration used to configure the authorization.

Default: **Derived from the certificate**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

Examples

Authorize synchronization in the current directory

This will add authorization for all certificates in the current directory to the given stations.

```
da.sync.authorize bnd
```

Add authorization

This will add authorization for any stations specified to accept the given digest for synchronization.

```
da.sync.authorize --digest=7058b93963e3d6bf1dbc22b88a1a1d4a1ebfefd0 bnd
```

File authorization

This will add authorization for any stations specified to accept the given certificate for synchronization.

```
da.sync.authorize --certificate=/tmp/sfa_sync.pem --deauthorize bnd
```

da.sync.client

This component synchronizes local data with data contained on a remote server. The specific data synchronized are determined by the profile selected. Synchronization can include both downloading new data from the server and uploading any local changes.

Usage

da.sync.client [switches...] [station]

Switches

--profile=STRING

This is the authorization profile to synchronize. The profile controls what data are synchronized with the remote server. Consult the station configuration and authorization for available profiles.

Default: **aerosol**

--quiet[=BOOLEAN**]**

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--update[=BOOLEAN**]**

When disabled, this causes the immediate data update to be skipped. The result is that the clean and averaged data are not updated until the next time the station tasks are run at the appropriate level. This normally means that the data will not be updated until the following night when the execution tasks run automatically.

Default: **Enabled**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

Examples

Run synchronization

This will run synchronization using the default profile.

```
da.sync.client bnd
```

da.sync.connect

This component synchronizes local data with data contained on a remote server. This synchronization server is set explicitly and the local data determined by the certificate that the server is using. Synchronization can include both downloading new data from the server and uploading any local changes.

Usage

da.sync.connect [switches...]

Switches

--port=**INTEGER**

This is the port on the remote server that the connection is established to.

Default: **14231**

This option only accepts integers greater than or equal to 1 and less than or equal to 65535.

--quiet[=**BOOLEAN]**

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--server=**STRING**

This is the hostname of the server that synchronization is performed with.

Default: **aero1.cmdl.noaa.gov**

--ssl-cert=**FILE**

This sets the certificate used for the connection.

Default: **sync.crt**

--ssl-key=**FILE**

This sets the key used for the connection.

Default: **sync.key**

--update[=BOOLEAN]

When disabled, this causes the immediate data update to be skipped. The result is that the clean and averaged data are not updated until the next time the station tasks are run at the appropriate level. This normally means that the data will not be updated until the following night when the execution tasks run automatically.

Default: **Enabled**

Examples

Run synchronization

This will run synchronization connecting to the default NOAA server.

```
da.sync.connect
```

da.sync.files

This component synchronizes local data with data contained on a remote server by using an exchange of static data files. The specific data synchronized are determined by the profile selected. Synchronization can involve creating files to transfer to the remote end as well as downloading updated files.

Usage

da.sync.files [switches...] [station...]

Switches

--after=TIME

This sets the time to synchronize after. This can be used to re-synchronize data after a given time or to completely resynchronize all data.

Default: **Last run time**

Undefined values are permitted, to indicate that the option is set but not to a specific time.

--download[=BOOLEAN]

This controls if data files containing remote changes are processed. When disabled any remote files are ignored and not processed. They may still be processed later if the synchronize is run without disabling them.

Default: **Enabled**

--profile=STRING...

This is the authorization profile to synchronize. The profile controls where data are read from and written to. Consult the station configuration and authorization for available profiles.

Default: **All available**

--quiet[=BOOLEAN]

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--update[=BOOLEAN]

When disabled, this causes the immediate data update to be skipped. The result is that the clean and averaged data are not updated until the next time the station tasks are run at the appropriate level. This normally means that the data will not be updated until the following night when the execution tasks run automatically.

Default: **Enabled**

--upload[=BOOLEAN]

This controls if data files containing local changes are generated. When disabled local changes are not uploaded.

Default: **Enabled**

--wait=NUMBER

This is the maximum time to wait for another process to complete, in seconds. If there are two concurrent processes for the same station and profile, the new one will wait at most this many seconds before giving up without performing an update.

Default: **Thirty seconds**

This option only accepts numbers greater than or equal to 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

Examples

Run synchronization

This will run synchronization using all available profiles.

```
da.sync.files
```

da.tap

This component provides a mechanism to create a tap in the data stream by writing a copy of the data seen to a file.

Usage

da.tap [switches...] [[station] variables times [archive]|[file]]

Switches

--file=FILE

This is the output file to write to. All data seen by this component is written to this file before being passed along.

Default: **tap.c3d**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data

are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively `-` (a single dash) may be used to explicitly specify standard input.

da.tasks

This component executes all periodic tasks on the system. Tasks are divided into integer priority levels with larger integers being run less frequently.

Usage

da.tasks [switches...] [station...]

Switches

--daily[=BOOLEAN]

This changes the execution mode such that the priority is increased when the current day is the first one of various intervals. That is, this executes at the specified priority when this is not the first day of something. It will execute at the priority plus 1000 when this is the first day of the week (defined as Saturday), plus 2000 when it is the first weekend of the month, and plus 3000 when it is the fist weekend of the year. When the priority is not set it defaults to 3000 when this option is used.

--detached[=BOOLEAN]

This option causes all tasks to be executed in detached processes. This allows for better isolation in the event of failure or other problems.

--priority=INTEGER

This is the maximum priority level of tasks to execute. Priority zero tasks are run every time data is processed. Priority 1000 is run "near real time" (generally every ten minutes). Priority 2000 is run "frequently" (generally hourly). Priority 3000 is run less frequently (generally daily). Priority 4000 is run on a more differed basis (generally weekly). Priority 5000 is run on a response time irrelevant basis (generally monthly). Priority 6000 is run on an even lower time basis (generally yearly).

Default: **0**

This option only accepts integers greater than or equal to 0.

--quiet[=BOOLEAN]

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

Examples

Run all immediate tasks

This will run all tasks that are flagged as immediate. Normally this is not required as it is run automatically by the data archive modifier.

```
da.tasks
```

Run all weekly tasks

This will run all tasks of weekly or higher frequency.

```
da.tasks --priority=4000
```

da.time.limit

This component provides a mechanism to apply limits to the times present in data. This includes absolute limiting as well as limiting based on distance from the start or end of the data present.

Usage

da.time.limit [switches...] [[station] variables times [archive]|[file]]

Switches

--end=TIME

This is the end time to output data. If this is set then no data after this time is output.

--exclude-end=INTERVAL

This if the maximum time accepted before the end of the last defined value time. That is, data that falls inside this interval from the end are rejected.

Default: **Unlimited**

The interval must be greater than zero in length.

--exclude-start=INTERVAL

This if the minimum time accepted after the start of the first defined value time. That is, data that falls inside this interval from the start are rejected.

Default: **Disabled**

The interval must be greater than zero in length.

--include-end=INTERVAL

This if the minimum time accepted before the end of the last defined value time. That is, data that falls outside this interval from the end are rejected.

Default: **Unlimited**

The interval must be greater than or equal to zero length.

--include-start=INTERVAL

This if the maximum time accepted after the start of the first defined value time. That is, data that falls outside this interval from the start are rejected.

Default: **Unlimited**

The interval must be greater than or equal to zero length.

--start=TIME

This is the start time to output data. If this is set then no data before this time is output.

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or

more variable specifications separated by , (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply BsB_S11 selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead ::BsB_S11:pm10 is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as bnd:BsB_S11,thd:BsB_S11 allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value everything can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as raw or clean_meta. The argument accepts multiple archives specified as regular expressions and separated by : or ; or ,. For example raw(_meta)? selects both the raw and raw_meta archives.

The special value allarchives may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

da.time.shift

This component provides a mechanism to apply an offset to the times in data.

Usage

da.time.shift [switches...] [[station] variables times [archive] | [file]]

Switches

--local[=BOOLEANS]

When enabled this causes data to be shifted as if the time stamps are in local time. This can be used to correct data that was mistakenly logged in local time. The computer time zone must match the logging time zone in this case. On Unix this can often be accomplished by setting the 'TZ' environment variable.

--shift=INTERVAL

This if the offset to shift data by. A positive offset shifts data forwards in time. Alignment (if specified) causes the start to be rounded down and the end to be rounded up for all data values. Invalid values (zero or data moved before the preceding values) are removed from the data stream.

Default: **None**

Arguments

If no bare word input specification is supplied then data are read from standard input.

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

file

This argument is used to specify the the file to read data from. If it is present and exists then data is read from the given file name instead of from standard input. Alternatively - (a single dash) may be used to explicitly specify standard input.

da.transfer.upload

This component uploads newly modified or generated data to a remote server. Generally, this action occurs periodically on remote acquisition systems to transfer the data they log to the main processing server.

Usage

da.transfer.upload [switches...] [station...]

Switches

--flush[=**BOOLEAN]**

When enabled, this causes a flush of the acquisition system before initiating the actual data transfer. This is used to attempt to transmit as much data as is available now instead of potentially still having some buffered for later transmission.

Default: **Disabled**

--profile=**STRING**

This is the profile name to execute. Multiple profiles can be defined to allow a single station to upload data to multiple destinations at potentially different intervals. Consult the station configuration for available profiles.

Default: **aerosol**

--quiet[=**BOOLEAN]**

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--wait=**NUMBER**

This is the maximum time to wait for another upload process to complete, in seconds. If there are two concurrent processes for the same station and profile, the new one will wait at most this many seconds before giving up without uploading new data. If this is set to undefined, it will wait forever for the other process to complete.

Default: **30 seconds**

This option only accepts numbers greater than or equal to 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

Examples

Upload all pending aerosol data

This will upload all pending data in the "aerosol" profile. This will not force the acquisition system (if any) to flush data, so the total upload may be slightly behind real time (the data will still be uploaded later once it is written).

```
da.transfer.upload
```

da.update.passed

This component sends updates the archive with data that has been passed by the station mentor. It updated any data that has been flagged as passed QA/QC but has not yet been inserted into the archive.

Usage

da.update.passed [switches...] [station...] [times]

Switches

--after=TIME

This sets the time to re-pass after. When set, this will update all data passed after the given time. This can be used to ignore the record of the last update, but doing so can cause there to be passed data that does not exist in the final archive. No effect when used with an explicit time range.

Default: **Last run time**

Undefined values are permitted, to indicate that the option is set but not to a specific time.

--detached[=BOOLEAN]

This option causes the update to be executed in detached processes. This allows for better isolation in the event of failure or other problems.

--fragment=INTERVAL

This sets the maximum time that an individual fragment of data is updated at. Any intervals longer than this threshold will be cut down to be shorter. This is intended to prevent excessively long archive access.

Default: **31 days**

Undefined intervals are accepted. The interval must be greater than zero in length.

--profile=**STRING...**

This sets the profiles to update. If set then only the profiles listed here will be considered for updating. Consult the station editing configuration for available profiles.

Default: **All passed profiles and the aerosol profile**

--quiet[=**BOOLEAN]**

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--wait=**NUMBER**

This is the maximum time to wait for another process to complete, in seconds. If there are two concurrent processes for the same station and profile, the new one will wait at most this many seconds before giving up without performing an update.

Default: **Thirty seconds**

This option only accepts numbers greater than or equal to 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

Examples

Update all stations

This will update passed data for all profiles and all stations.

```
da.update.passed 2015-05-01 2015-05-03
```

da.update.plots

This component updates the fixed output plots generate from the data. Normally this consists of a set of plots intended for public display from the station. These plots cover data from the latest available moving backwards, possibly to the start of data entirely.

Usage

da.update.plots [switches...] [station...]

Switches

--profile=**STRING...**

This sets the profiles to update. If set then only the profiles listed here will be considered for updating. Consult the station configuration for available profiles.

Default: **All available profiles**

--quiet[=**BOOLEAN]**

If set then no progress output is displayed and any confirmation prompts are bypassed.

Default: **Disabled**

--recalculate[=**BOOLEAN]**

If set then the plot data bounds are recalculated from scratch. This can be used if the plots need to be regenerated aftercorrupted data has been removed.

--reprocess[=**BOOLEAN]**

If set then all plots are regenerated, not just those that have been updated since the last run.

--wait=**NUMBER**

This is the maximum time to wait for another process to complete, in seconds. If there are two concurrent processes for the same station and profile, the new one will wait at most this many seconds before giving up without performing an update.

Default: **Thirty seconds**

This option only accepts numbers greater than or equal to 0. Undefined values are permitted, to indicate that the option is set but not to a specific value.

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code

of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

Examples

Update all stations

This will update plots for all profiles and all stations.

```
da.update.plots
```

da.config.text

This program provides an interface to display or edit configuration information as text files. This generally allows for a flat file representation similar to a classif configuration file.

Usage

da.config.text [switches...] [station] variables [times] [archive]

Switches

--editor=**STRING**

This is the editor command that will be invoked when in read→modify→write editing mode. If not set then the editor will be selected by what is set in the $EDITOR environment variable. If that is not set then an editor will be selected based on the presence of a GUI environment.

Default: **Automatically selected**

--file=**FILE**

This sets the file name to read or write to. By default standard input or output is used.

Default: **Disabled, used standard input/output**

--flat[=**BOOLEAN]**

When set, this outputs on a single data indentifer and time bound to be output, instead of the full specification.

--mode=**TYPE**

This sets the mode the flat file generator works in.

Default: **Read data from the archive**

The possible values are:

--mode=datainput

Read data from the archive or standard input and generate the flat file as output.

--mode=dataoutput

Read and parse data from the file and generate data on standard output.

--mode=edit

Read data from the archive to a temporary file then open an editor on that file. Once the editor is closed write the changes back to the archive.

--mode=modify

Same as --mode=**edit**

--mode=read

Read data from the archive and generate a flat file as output.

--mode=write

Write data to the archive from an input flag file.

--mode=xml

Read data from the archive to a temporary file in XML then open an editor on that file. Once the editor is closed write the changes back to the archive.

--output-bounds[=BOOLEAN]

When set, this causes the data bounds to always be output instead of only output when needed.

--output-identifiers[=BOOLEAN]

When set, this causes the data identifiers (station, archive, variable, and flavors) to always be output instead of only output when needed.

--read=TYPE

This sets the mode data are read with. This is used to control what data are displayed for editing.

Default: **Accept default metadata**

The possible values are:

--read=all

Same as --read=**everything**

--read=everything

Accept all input data.

--read=metadata

Accept only metadata from the default archive. All read default data are ignored.

--read=nodefault

Do not accept any default station data or metadata.

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

variables

This argument may be split into multiple actual program arguments. Each part consists of one or more variable specifications separated by `,` (commas). The station and archive used if none are explicitly given are defined in the other arguments or inferred from system defaults. For example simply `BsB_S11` selects the blue scattering from the S11 instrument and returns all cut sizes and metadata. If instead `::BsB_S11:pm10` is used then only PM10 scattering is returned. That is, the selection specifies "any" station and archive (the defaults are set in the other arguments) then requires that it has the "PM10" flavor. A specification such as `bnd:BsB_S11,thd:BsB_S11` allows for data selection from multiple stations.

This may also be used to select a "record" type alias such as "S11a". In this mode all the variables defined in the alias are returned.

Finally the special value `everything` can be used to select all available data for the given station(s) and archive(s).

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

If the mode is set to read data from standard input, no bare word arguments are required.

da.output.cpd2

This program generates CPD2 style records by reading the CPD3 archive directly. This is normally only used as part of the legacy interface to work with old CPD2 programs.

Usage

da.output.cpd2 [station] [records] times [archive]

Arguments

station

This argument is used to specify the station used to look up variables that do not include a station as part of an archive read specification. The station is the three letter GAW station code of the location, such as `BND`. The argument accepts multiple stations specified as regular expressions and separated by `:` or `;` or `,`. For example `BND,MLO` and `(BND|MLO)` are two ways of selecting both the *BND* and *MLO* stations.

The special value `allstations` may also be specified to select all stations.

records

This argument is a list of CPD2 record names, separated by `:` or `;` or `,` to be generated. For example `S11a` for the **S11** instrument **a** record type. Regular expressions are accepted but note that the stations specification is evaluated first, so they must not match both if the station is omitted. The special value `everything` is accepted to generate data for all available records.

times

This argument may be split into multiple actual program arguments. This full list of time arguments defines a time bounds list that sets the range of data queried. The final resulting data are clipped to the given time bounds but any values that intersect them are returned.

archive

This argument is used to specify the archive used to look up variables that do not include an archive as part of an archive read specification. The archive is the internal name, such as `raw` or `clean_meta`. The argument accepts multiple archives specified as regular expressions and separated by `:` or `;` or `,`. For example `raw(_meta)?` selects both the `raw` and `raw_meta` archives.

The special value `allarchives` may also be used to select all available archives.

Examples

General usage

This generates CPD2 style **S11a** records for **BND**.

```
da.output.cpd2 bnd S11a 2016-05-01 1d
```

da.sync.peer

This program performs bi-directional synchronization of data with another peer. The connection is normally tunneled over another transport mechanism to provide security.

Usage

da.sync.peer [switches...] [station]

Switches

--editor=**STRING**

This is the editor command that will be invoked when in read→modify→write editing mode. If not set then the editor will be selected by what is set in the $EDITOR environment variable. If that is not set then an editor will be selected based on the presence of a GUI environment.

Default: **Automatically selected**

--file=**FILE**

This sets the file name to read or write to. By default standard input or output is used.

Default: **Disabled, used standard input/output**

--flat[=**BOOLEAN]**

When set, this outputs on a single data indentifer and time bound to be output, instead of the full specification.

--mode=**TYPE**

This sets the mode the flat file generator works in.

Default: **Read data from the archive**

The possible values are:

--mode=datainput

Read data from the archive or standard input and generate the flat file as output.

--mode=dataoutput

Read and parse data from the file and generate data on standard output.

--mode=edit

Read data from the archive to a temporary file then open an editor on that file. Once the editor is closed write the changes back to the archive.

--mode=modify

Same as --mode=**edit**

--mode=read

Read data from the archive and generate a flat file as output.

--mode=write

Write data to the archive from an input flag file.

--mode=xml

Read data from the archive to a temporary file in XML then open an editor on that file. Once the editor is closed write the changes back to the archive.

--output-bounds[=BOOLEAN]

When set, this causes the data bounds to always be output instead of only output when needed.

--output-identifiers[=BOOLEAN]

When set, this causes the data identifiers (station, archive, variable, and flavors) to always be output instead of only output when needed.

--read=TYPE

This sets the mode data are read with. This is used to control what data are displayed for editing.

Default: **Accept default metadata**

The possible values are:

--read=all

Same as --read=**everything**

--read=everything

Accept all input data.

--read=metadata

Accept only metadata from the default archive. All read default data are ignored.

--read=nodefault

Do not accept any default station data or metadata.

Arguments

station

This argument is used to specify the station that synchronization is attempted for. It defines both the data requested from the remote peer and the local data accepted. The station is the three letter GAW station code of the location, such as BND.

da.sync.read

This program reads a synchronization stream created with da.sync.write and applies the changes to the local archive. It is used with manual transfer of that stream to accomplish synchronization of two archives.

Usage

da.sync.read [switches...] [station] [file]

Switches

--editor=STRING

This is the editor command that will be invoked when in read→modify→write editing mode. If not set then the editor will be selected by what is set in the $EDITOR environment variable. If that is not set then an editor will be selected based on the presence of a GUI environment.

Default: **Automatically selected**

--file=FILE

This sets the file name to read or write to. By default standard input or output is used.

Default: **Disabled, used standard input/output**

--flat[=BOOLEAN**]**

When set, this outputs on a single data indentifer and time bound to be output, instead of the full specification.

--mode=TYPE

This sets the mode the flat file generator works in.

Default: **Read data from the archive**

The possible values are:

--mode=datainput

Read data from the archive or standard input and generate the flat file as output.

--mode=dataoutput

Read and parse data from the file and generate data on standard output.

--mode=edit

Read data from the archive to a temporary file then open an editor on that file. Once the editor is closed write the changes back to the archive.

--mode=modify

Same as --mode=**edit**

--mode=read

Read data from the archive and generate a flat file as output.

--mode=write

Write data to the archive from an input flag file.

--mode=xml

Read data from the archive to a temporary file in XML then open an editor on that file. Once the editor is closed write the changes back to the archive.

--output-bounds[=BOOLEAN**]**

When set, this causes the data bounds to always be output instead of only output when needed.

--output-identifiers[=BOOLEAN]

When set, this causes the data identifiers (station, archive, variable, and flavors) to always be output instead of only output when needed.

--read=TYPE

This sets the mode data are read with. This is used to control what data are displayed for editing.

Default: **Accept default metadata**

The possible values are:

--read=all

Same as --read=**everything**

--read=everything

Accept all input data.

--read=metadata

Accept only metadata from the default archive. All read default data are ignored.

--read=nodefault

Do not accept any default station data or metadata.

Arguments

station

This argument is used to specify the station that synchronization is accepted for. The incoming data must match the station and profile to be accepted. The station is the three letter GAW station code of the location, such as BND.

file

This argument is used to specify the the file to read data from. If it is absent or set to - (a single dash) then the stream is read from standard input.

da.sync.write

This program examines the archive for changes and writes them to a specialized output stream. That output stream is then used in conjunction with da.sync.read to perform manual synchronization.

Usage

da.sync.write [switches...] [station] [file]

Switches

--editor=STRING

This is the editor command that will be invoked when in read→modify→write editing mode. If not set then the editor will be selected by what is set in the $EDITOR environment variable. If that is not set then an editor will be selected based on the presence of a GUI environment.

Default: **Automatically selected**

--file=FILE

This sets the file name to read or write to. By default standard input or output is used.

Default: **Disabled, used standard input/output**

--flat[=BOOLEAN**]**

When set, this outputs on a single data indentifer and time bound to be output, instead of the full specification.

--mode=TYPE

This sets the mode the flat file generator works in.

Default: **Read data from the archive**

The possible values are:

--mode=datainput

Read data from the archive or standard input and generate the flat file as output.

--mode=dataoutput

Read and parse data from the file and generate data on standard output.

--mode=edit

Read data from the archive to a temporary file then open an editor on that file. Once the editor is closed write the changes back to the archive.

--mode=modify

Same as --mode=**edit**

--mode=read

Read data from the archive and generate a flat file as output.

--mode=write

Write data to the archive from an input flag file.

--mode=xml

Read data from the archive to a temporary file in XML then open an editor on that file. Once the editor is closed write the changes back to the archive.

--output-bounds[=BOOLEAN**]**

When set, this causes the data bounds to always be output instead of only output when needed.

--output-identifiers[=BOOLEAN]

When set, this causes the data identifiers (station, archive, variable, and flavors) to always be output instead of only output when needed.

--read=TYPE

This sets the mode data are read with. This is used to control what data are displayed for editing.

Default: **Accept default metadata**

The possible values are:

--read=all

Same as --read=**everything**

--read=everything

Accept all input data.

--read=metadata

Accept only metadata from the default archive. All read default data are ignored.

--read=nodefault

Do not accept any default station data or metadata.

Arguments

station

This argument is used to specify the station that synchronization data are written out for. The station is the three letter GAW station code of the location, such as `BND`.

file

This argument is used to specify the the file to write data to. If it is absent or set to `-` (a single dash) then the stream is written to standard output.

Option Types

Boolean Option

Boolean options are switches that accept a true or false value. When the switch is set without an equals, it sets the option to true.

Option turned on

```
--switch
```

If the switch contains an equals, the value controls the enabled state of the option.

Enabled values

- Any non-zero integer
- The word `on`
- The word `yes`
- The word `true`

Disabled values

- The integer zero
- The word `off`
- The word `no`
- The word `false`

That is, the option can be toggled by setting it to integers (with zero being off), by setting it to "yes" or "no", or by setting it to "true" or "false". The word specifications are not case sensitive.

Option explicitly turned on

```
--switch=1
--switch=on
--switch=yes
--switch=TRUE
```

Option explicitly turned off

```
--switch=0
--switch=off
--switch=NO
--switch=false
```

Single Integer Option

Integer options accept a single base ten integer. Leading zeros are ignored and only decimal digits are accepted. Commas or place delimiters are not supported.

The accepted range and interpretation depends on the context of the option. If a lower bound is defined the value must be greater than or equal to the lower limit. Likewise, if an upper bound is defined, the value must be less than or equal to it.

Option set to a finite number

```
--switch=0
--switch=20
--switch=-30
```

In some contexts integer options may accept undefined values. This indicates to the program that the option was set by the caller but not to a specific value. This is commonly used to indicate an explicitly infinite value.

Undefined values

- An empty value
- The word `undef`
- The word `undefined`
- The word `inf`
- The word `infinity`
- The word `mvc`

Option set to undefined

```
--switch
--switch=undef
--switch=Undefined
--switch=inf
--switch=INFINITY
--switch=mvc
```

Single Number Option

Real number options accept base ten numbers with optional fractional components. Additionally, leading zeros are ignored and numbers can be specified in scientific notation using 'e' syntax. Commas or place delimiters are not supported.

The accepted range and interpretation depends on the context of the option.

Option set to a finite number

```
--switch=0
--switch=20.5
--switch=-3E-1
```

In some contexts number options may accept undefined values. This indicates to the program that the option was set by the caller but not to a specific value. This is commonly used to indicate an explicitly infinite value.

Undefined values

- An empty value
- The word `undef`
- The word `undefined`
- The word `inf`
- The word `infinity`
- The word `mvc`

Option set to undefined

```
--switch
--switch=undef
--switch=Undefined
--switch=inf
--switch=INFINITY
--switch=mvc
```

String Option

String options accept a string of text. The interpretation of the text depends on the option.

Simple string

```
--switch=SomeValue
```

String with spaces

```
--switch='Some other value'
```

Enumeration Option

Enumeration options accept a single value from a set of possible ones. Values are specified using a string of text and are not case sensitive.

For example a mode option might allow the values `instrument` and `variable` so exactly one of them

would be valid.

Set value

```
--switch=value
```

Some values may also be aliases for another one. In this mode specifying the alias value has the same effect as specifying the main one. For example if the value `Excel` is an alias for `xl` then either one can be used and it will have the same effect.

Instrument Suffix Option

Instrument suffix options specify a list of instrument suffixes to operate on. An instrument suffix is the final part after the underscore in variable names. For example *S11* is usually the primary nephelometer. Multiple suffixes are specified by separating them with `,` or `;` or `:` and empty ones are ignored.

Option set

```
--switch=S11
--switch=S11,S12
--switch=A11:A12
```

The option may also be set an empty list of instruments by not specifying a value.

Option set to empty

```
--switch
--switch=
```

Calibration Option

Calibration options specify a polynomial applied to the input to produce a calibrated output. The most common usages are conversions from raw values (e.g. voltages) to logical physical units and the correct for sensor specific anomalies. The calibration is specified as a list of numeric coefficients in ascending power order, starting from the constant (x^0). Each coefficient is a base ten decimal number with an optional fractional part. Additionally, leading zeros are ignored and numbers can be specified in scientific notation using 'e' syntax.

Coefficients are are delimited using `,` or `;` or `:` with empty fields interpreted as zeros.

Identity

```
--switch=0,1
--switch=:1.0
```

Third order polynomial

```
--switch=0,1,2
--switch=-1E1:-2:0.5
```

Depending on the context the polynomial may also be replaced with just a constant. This means that the result is always a constant value, regardless of what the input value is.

Constant polynomial

```
--switch=0.5
```

Additionally some contexts accept invalid (empty) polynomials. The result of an invalid polynomial is always an undefined value. This is used to specify that the result is never a valid number.

Invalid polynomial

```
--switch=
--switch
```

Variable Selection Option

Variable selection options accept a list of variables separated by , or ; and will match if any selection in the list applies. The individual selections can themselves be regular expressions that match more than a single variable.

Simple variable

```
--switch=BsG_S11
--switch=Bs[BGR]_S11
```

Multiple expressions

```
--switch=BsB_S11,BsG_S11
--switch=::BsB_S11:pm1,::BsG_S11:pm10
```

Input Value Option

Input value options are options that specify a real numbers used as an input to a calculation. In their simplest form they are simply constant values specified in base ten. Leading zeros are ignored and numbers can be specified in scientific notation using 'e' syntax.

Constant value

```
--switch=1.0
--switch=1.2E3
```

In more complex forms, the option is split into fields separated by `,` and the meaning is determined by the first field. If the first field is a variable specification, the input is based on the value of that variable in the data. The second field is a calibration applied to it and the third is the "fallback" constant value used when the value is not valid in data.

Variable specification

```
--switch=BsB_S11
--switch=T_S11,273.15:1.0,280
```

> **NOTE**
>
> *Multiple variables matched*
>
> If multiple variables match the variable the order they are check in is not defined. That is, the input will take an arbitrary one of them if multiple have a valid value.

If the option contains a `/` then it is treated as time dependent. This means that all fields after the value ones (the constant or variable, etc) are part of a time specification. The time specification determines the time range the value is applied to.

Time dependency

```
--switch=BsB_S11,,,2016:10/BsB_S11,-2.0,,,2016:10
--switch=1.0,2015,2016/2.0,2016
```

Data Operation Option

Data operation options select parameters in the data to perform an operation on. In the simplest form this consists of just a variable name specification. This results in operating on all values matching the variable name.

Simple variable

```
--switch=BsG_S11
```

However, the full data selection syntax is also supported.

If the option contains a `/` then it is treated as time dependent. In this context the value is split into fields separated by `,` with the first field being the selection and all further ones specifying the time range. This means that the selection in the first field is applied for the time range specified in subsequent ones.

Time dependency

```
--switch=BsB_S11,,2016:10/BsB_S11,2016:10
--switch='BsB_S11;BsG_S11,,2016/Bs[RB]_S11,2016'
```

Interval Selection Option

Interval selection options specify an interval of time. This is used as an offset from one time to another but often this defines a block of time. For example, the interval applied to a start time gives an end time and the two together determine the interval being specified.

In the simplest form, the option is just a single time interval.

Simple interval

```
--switch=1h
```

Depending on the context the option may also permit zero and/or negative intervals.

If the option contains a `/` then it is treated as time dependent. In this context the value is split into fields separated by `,` with the first field being the interval and all further ones specifying the time range. This means that the selection in the first field is applied for the time range specified in subsequent ones.

Time dependency

```
--switch=1h,,2016:10/2h,2016:10
--switch='0,,2016/-1 minute aligned,2016'
```

If the option permits an undefined interval, then the interval can also be one of:

`undef`
`undefined`
`none`

When one of these is used, the result of the offset is an undefined time. The interpretation of undefined times depends on the context, but it is often "infinite" or "forever".

Time Offset Option

Time offset options accept a single time interval. The offset may be restricted to being non-zero and/or only positive.

Simple interval

```
--switch=1h
```

Depending on the context the option may also permit zero intervals. Similarly it may accept an undefined interval:

`undef`
`undefined`
`none`

When one of these is used, the result of the offset is an undefined time.

Undefined interval

```
--switch=undef
--switch=undefined
```

Time Option

Time options accept a single specification of a point in time. Take care with time specifications that contain special characters that may need escaping or quoting.

Single time

```
--switch=2016-05-01
--switch='2016-05-01T00:00:00Z'
```

In some contexts the option may also accept infinite or undefined times.

Undefined time

```
--switch
--switch=
--switch=infinity
```

Multiple Unique Number Option

Multiple real number options accept base ten numbers with optional fractional components. Additionally, leading zeros are ignored and numbers can be specified in scientific notation using 'e' syntax. Multiple numbers are delimited using `,` or `;` or `:` with duplicates ignored.

The accepted range and interpretation depends on the context of the option.

Option set

```
--switch=0
--switch=1,2,3
--switch=1E3:3.0
```

The option may also be set an empty set by not specifying a value.

Option set to empty

```
--switch
--switch=
```

Multiple Ordered Number Option

Multiple real number options accept base ten numbers with optional fractional components. Additionally, leading zeros are ignored and numbers can be specified in scientific notation using 'e' syntax. Multiple numbers are delimited using `,` or `;` or `:` and the order they are specified in is perserved.

Option set

```
--switch=0
--switch=1,2,3
--switch=1E3:3.0
```

The option may also be set an empty list by not specifying a value.

Option set to empty

```
--switch
--switch=
```

Multiple Unique String Option

Multiple string options accept a list of unique strings of text. Multiple strings are specified by separating them with `,` or `;` or `:` and empty ones are ignored.

Option set

```
--switch=Text
--switch=First,Second
--switch=1:2:Three
```

The option may also be set an empty list of strings by not specifying a value.

Option set to empty

```
--switch
--switch=
```

Interval Block Option

Interval selection options specify a positive an interval of time. Time time interval must be greater than or equal to one of whichever unit is selected.

Simple interval

```
--switch=1h
```

File Option

File options are switches that accept a file name to operate on. Depending on the context this file may be read from or written to. In general, read operations require the file to exist before the program is started. Both relative and absolute paths are accepted. Case sensitivity depends on the underlying filesystem, but in general it should be assumed.

Absolute path

```
--switch=/path/to/file.dat
```

Relative path

```
--switch=file.dat
```

Directory Option

Directory options are switches that accept a directory path. The most common use is to specify a directory to output file in. In this use, the file names are generated by some internal procedure and written to the specified directory. In general, the directory should exist before the program is started. Both relative and absolute path names are accepted. Case sensitivity depends on the underlying filesystem, but in general it should be assumed.

Absolute directory

```
--switch=/path/to/directory
```

Relative directory

```
--switch=local/directory
```

Script Code Option

Script options accept CPD3 style script code to be executed. The context and availability of features depends on the option. CPD3 script code is derived from Lua 5.1 with CPD3 and LuaJIT extensions. Script code is often executed in a sandboxed environment, meaning that many OS level functions are not available. Most script print output is directed to the debug stream, and so is not visible unless explicitly enabled.

Basic script

```
--switch='for i=1,10 do print("Value = " .. i); end'
```

Script with stream context

```
--switch='for v in data do v.START = v.START + 100; end'
```

In contexts where the script is expected to produce a value, it should return that value as if called in a Lua function.

Script value result

```
--switch='return data.BsG_S11 == 0'
```

Smoother Option

Smoother options specify a smoother applied to the input sequence. This smoother may be used for value smoother, spike detection and/or stability detection.

In the simplest form, the option is just a single smoother specification. The smoother specification consists of a sequence of fields separated by `,` with the meaning and number determined by the first field. The type specification is not case sensitive:

`disable`
: Disable smoothing. This is always stable without spikes but never gives a valid value.

`last`
: Return the last valid value. This is always stable without spikes and returns the last valid value seen.

`anylast`
: Return the latest value even if it is invalid. This is always stable without spikes and returns the last value even if it was invalid.

`1p`, `lowpass`, `1plp`, `singlepole`, **or** `singlepolelowpass`
: A single pole low pass digital filter. This is a simple infinite response filter. The time constant is the second field and is specified as a time interval that is the nominal time the filter takes to reach 63.2% of its final value after a step change. The third field is a time interval specifying the maximum time between points before it is considered a gap in data and the filter is reset. The fourth field is a boolean value (`on`, `yes`, `true` or none zero vs `off`, `no`, `false` or zero) indicating whether or not the filter should reset after an undefined value. The filter is considered stable if the most recent value is within the band set by the fifth field. The filter reports a spike if the most recent value falls outside the band defined by the sixth field.

`4p`, `4plp`, `fourpole`, **or** `fourpolelowpass`
: A four pole low pass digital filter. This is a simple infinite response filter. The time constant is the second field and is specified as a time interval that is the nominal time the filter takes to reach 63.2% of its final value after a step change. The third field is a time interval specifying the maximum time between points before it is considered a gap in data and the filter is reset. The fourth field is a boolean value (`on`, `yes`, `true` or none zero vs `off`, `no`, `false` or zero) indicating whether or not the filter should reset after an undefined value. The filter is considered stable if

the most recent value is within the band set by the fifth field. The filter reports a spike if the most recent value falls outside the band defined by the sixth field.

A valid base ten decimal number

A finite length mean smoother of the specified minimum length in seconds. The second field specifies the maximum length of the smoother. The third field specifies the number of seconds to discard on initialization before starting smoothing. The fourth field specifies the maximum relative standard deviation for the smoother to be considered stable. The filter reports a spike if the most recent value is outside the band defined by the sixth field.

Three minute single pole low pass

```
--switch=1p,3m
```

30 second minimum and 60 second maximum mean

```
--switch=30,60,10,0.01,2.0
```

If the option contains a `/` then it is treated as time dependent. In this context the value is split into fields separated by `,` with the first field being the interval and all further ones specifying the time range. The first fields determine the smoother and the remaining fields set the time range specified that type of smoother applies to.

Time dependency

```
--switch=30,60,10,0.01,2.0,,2016:10/90,,,,,2016:10
```

Band Specification

A band is the fraction minus one relative to zero. So a band of 2.0 sets the valid range from 1/3 below to three times above. That is, the band minus one is multiplied or divided by the relative value to get the absolute bounds.

SSL Specification Option

SSL options specify a key and/or certificate for SSL encryption and authentication. The usage of the key and certificate depends on the context and one may be optional. Both the key and certificate are given as either paths to files or directly as PEM specifications. The components are separated by `:` or `;` or `,` or `|` with the key being the first field and the certificate the second.

Key and certificate files

```
--switch=key.pem,cert.pem
```

Time Dependent String Option

Time dependent string options accept a string of text with optional variablity in time. In the simplest form this is just a constant string for all time.

Constant text

```
--switch=Value
```

If the option contains a `/` then it is treated as time dependent. In this context the value is split into fields separated by `,` with the first field being the interval and all further ones specifying the time range. This means that the string in the first field is the value for the time range specified in subsequent ones.

Time dependency

```
--switch=First,,2016:10/second,2016:10
--switch='Zero,,2016/One,2016'
```

Time Dependent Boolean Option

Time dependent boolean options accept true or false values that optionally vary in time. When the switch is set without an equals, it sets the option to true.

Option turned on

```
--switch
```

If the switch contains an equals, the value controls the enabled state of the option.

Enabled values

- Any non-zero integer
- The word `on`
- The word `yes`
- The word `true`

Disabled values

- The integer zero
- The word `off`
- The word `no`
- The word `false`

That is, the option can be toggled by setting it to integers (with zero being off), by setting it to "yes" or "no", or by setting it to "true" or "false". The word specifications are not case sensitive.

Option explicitly turned on

```
--switch=1
--switch=on
--switch=yes
--switch=TRUE
```

Option explicitly turned off

```
--switch=0
--switch=off
--switch=NO
--switch=false
```

If the option contains a `/` then it is treated as time dependent. In this context the value is split into fields separated by `,` with the first field being the interval and all further ones specifying the time range. This means that the truth value in the first field is the value for the time range specified in subsequent ones.

Time dependency

```
--switch=true,,2016:10/false,2016:10
```

Time Dependent Number Option

Time dependent real number options accept base ten numbers with optional fractional components. Additionally, leading zeros are ignored and numbers can be specified in scientific notation using 'e' syntax. Commas or place delimiters are not supported. In the simplest form this is just a constant value for all time.

The accepted range and interpretation depends on the context of the option.

Option set to a finite number

```
--switch=0
--switch=20.5
--switch=-3E-1
```

In some contexts number options may accept undefined values. This indicates to the program that the option was set by the caller but not to a specific value. This is commonly used to indicate an explicitly infinite value.

Undefined values

- An empty value
- The word `undef`
- The word `undefined`

- The word `inf`
- The word `infinity`
- The word `mvc`

Option set to undefined

```
--switch
--switch=undef
--switch=Undefined
--switch=inf
--switch=INFINITY
--switch=mvc
```

If the option contains a `/` then it is treated as time dependent. In this context the value is split into fields separated by `,` with the first field being the interval and all further ones specifying the time range. This means that the number in the first field is the value for the time range specified in subsequent ones.

Time dependency

```
--switch=1.0,,2016:10/2.0,2016:10,
--switch='23,,2016/-1E3,2016'
```

Time Dependent Integer Option

Time dependent integer options accept a single base ten integer. Leading zeros are ignored and only decimal digits are accepted. Commas or place delimiters are not supported. In the simplest form this is just a constant value for all time.

The accepted range and interpretation depends on the context of the option. If a lower bound is defined the value must be greater than or equal to the lower limit. Likewise, if an upper bound is defined, the value must be less than or equal to it.

Option set to a finite number

```
--switch=0
--switch=20
--switch=-30
```

In some contexts integer options may accept undefined values. This indicates to the program that the option was set by the caller but not to a specific value. This is commonly used to indicate an explicitly infinite value.

Undefined values

- An empty value
- The word `undef`

- The word `undefined`
- The word `inf`
- The word `infinity`
- The word `mvc`

Option set to undefined

```
--switch
--switch=undef
--switch=Undefined
--switch=inf
--switch=INFINITY
--switch=mvc
```

If the option contains a `/` then it is treated as time dependent. In this context the value is split into fields separated by `,` with the first field being the interval and all further ones specifying the time range. This means that the number in the first field is the value for the time range specified in subsequent ones.

Time dependency

```
--switch=1,,2016:10/2,2016:10,
--switch='23,,2016/-10,2016'
```

Time Dependent Calibration Option

Time dependent calibration options specify a polynomial applied to the input to produce a calibrated output with the polynomial being optionally variable in time. The calibration is specified as a list of numeric coefficients in ascending power order, starting from the constant (x^0). Each coefficient is a base ten decimal number with an optional fractional part. Additionally, leading zeros are ignored and numbers can be specified in scientific notation using 'e' syntax.

Coefficients are are delimited using `,` or `;` or `:` with empty fields interpreted as zeros.

Identity

```
--switch=0,1
--switch=:1.0
```

Third order polynomial

```
--switch=0,1,2
--switch=-1E1:-2:0.5
```

If the option contains a `/` then it is treated as time dependent. In this context the value is split into fields separated by `,` with the first field being the interval and all further ones specifying the time

range. This means that the coefficients in the first field are the polynomial for the time range specified in subsequent ones.

Time dependency

```
--switch=0:1,,2016:10/-0.5:2.0,2016:10,
```

Common Formats

System Data Components

The system breaks data specifications down into several components. These components are generally used to specify data requested or accessed.

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 heirarchy 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 1µm diameter. Flavors are **not** case sensitive.

Some common flavors include:

PM1

Data have been size selected to less than 1µm diameter.

PM25

Data have been size selected to less than 2.5µm diameter.

PM10

Data have been size selected to less than 10µm 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.

Data Parameter Specification

In general a variable argument specification consists of one or more fields separated by `:` with the number of fields determining their interpretation.

Empty fields are treated as their defaults or matching anything, depending on the context.

All fields are considered as regular expressions to be matched against the whole field. In general this means that the simple case of specifying a whole field literally will simply work. It also allows for more complex specifications using special characters. The regular expression syntax is similar to Perl's form with some of the more advanced forms (e.g. negative lookahead) not supported. When using regular expressions, be sure that they are properly escaped if passed from a command line shell.

One field

When only a single field is present it is interpreted as a variable name. The station, archive, and flavors matching are left at their default values. For example `BsG_S11` selects only green scattering from the S11 instrument.

Two fields

When the specification consists of two fields, they specify the archive and variable in order. For example `raw:BsG_S11` selects the green scattering from the `raw` archive.

Three or more fields

With three fields the first three are the station, archive, and variable, in order. For example `bnd:raw:BsG_S11` selects the green scattering from the `raw` archive for the `BND` station.

Any remaining fields after the first three form the flavors selection. Each field in a flavors selection specifies a single flavor to match. The interpretation depends on the first character of the field:

No special character or a `+`

The field is added to the list of flavors required to be present. For example `pm1` and `+pm1` both require the `PM1` flavor.

Starting with a `!` or `-`

The field is added to the list of flavors required to be absent. For example `!pm10` and `-pm10` both exclude data with the `PM10` flavor.

Starting with a `=`

The field is added to the exact flavors specification. Any required or excluded flavors are ignored. The exact flavors specification gives the full list of flavors matched exactly. For example `=pm25` selects only data with just the `PM25` flavor and ignores all other data.

Example 1. Selecting a single variable

```
BsG_S11
```

Example 2. Selecting multiple variables with a regular expression

```
Bs[BGR]_S11
```

Example 3. Selecting only PM10 data

```
::BsG_S11:pm10
```

Example 4. Selecting metadata without a size selection

```
:.*_meta::-pm1:-pm10:-pm25
```

Example 5. Selecting only values with no flavors (size, coverage, etc)

```
(brw|smo)::BsG_S11:=
```

The meaning of the individual components is described above.

Single Time

In general a time specification accepts most ISO8601 formats in addition to a number of auxiliary formats.

The literal `now`

The current time.

An infinite bound

This is one of the following forms:

Empty

All space or zero length.

`0`

The number zero.

`none`

`undef` **or** `undefined`

`inf` **or** `infinity`

`all`

`forever`

An integer and an offset multiplier

This offsets the time in the other reference point. It is only valid when a reference point exists, such as valid other bound in a pair of times. The form is of "X<unit>" where "X" is a non-negative number, and "<unit>" is one of:

`msec` **or** `milliseconds`

`s`, `sec`, **or** `seconds`

`m`, `min`, **or** `minutes`

Exactly X * 60 seconds.

`h` **or** `hours`

Exactly X * 3600 seconds.

`d` **or** `days`

Exactly X * 86400 seconds.

`w` **or** `weeks`

Exactly X * 604800 seconds.

`mo`, `mon`, **or** `months`

Preserves the time of day and day of month.

`q`, `qtr`, **or** `quarter`

Preserves the time after the start of the quarter.

`y` **or** `years`

Preserves the month, day of month, and time of day.

Example 6. One week from the reference point

```
1w
```

The offset may also also be followed by `a` or `aligned` to align to the boundary the multiplier represents.

Example 7. Rounding to the quarter boundary

```
0qa
```

In general the count must be an integer. However, decimal numbers are accepted for the exact fixed length specifications and are interpreted as the largest whole unit below the specification that gives the exact time.

Example 8. 36 hours as a number of days

```
1.5d
```

Alignment is not allowed on fractional specifications. Without the alignment enabled, the count number must be positive (non-zero). If the count is omitted then the specification is for a single of the unit.

Date and/or time

If present the date must come first. The date must start with a four digit year, followed by an optional month and day of month. Each field must be separated by either a dash or a slash. The time then follows either a space separator or a `T`. Time is specified on a 24 hour clock, with the hours first, seconds being optional. Each field must be separated by a `:`. The specification may optionally end in a `Z`.

Example 9. ISO8601 specification

```
2010-03-10T00:15:00Z
```

Example 10. Excel-like specification

```
2010/03/10 00:15:00
```

Example 11. Date only

```
2010-03-10
```

Example 12. Date and time without delimiters

```
20100310T001500Z
```

Example 13. Time only

```
15:00:12.123Z
```

Example 14. Year only

2010

If only a time is specified, the date is inferred from the reference point such that it would be reasonable relative to it.

Example 15. Data inference

If the this is the start bound and the time is after the reference end bound, then it is assumed to be in the prior day

A date and/or time may optionally be followed by a time zone offset. This takes the form of plus or minus and a two or four digit integer or `hh:mm`. This number is subtracted or added to the given time to give UTC.

Year and week or week

One of the following forms:

YYYYwWW

YYYYwWW-D

Monday is day one.

wWW

wWW-D

Monday is day one.

Note that the "WW" component may be a single digit. The exact time that the week specifies depends on the reference point: if this is a start bound, then time specified is the start of that week, otherwise it is the end. If the year is omitted then it is inferred from the reference point as with only a time above.

Year and quarter or quarter

One of the following forms:

YYYYqQ

qQ

The exact time that the quarter specifies depends on the reference point as above with weeks.

Year and DOY or DOY

A four digit year separated from a DOY by any of `:`, `;`, `,` or space(s). The day of year starts with 1.0 being midnight January 1st of that year. The separator may also be omitted or replaced with `-` if there are exactly three digits in the integer component of the DOY. If the DOY has exactly two decimal digits, it is assumed to specify an hour and rounded to the nearest hour boundary.

Example 16. Midnight January 1st, 2010

```
2010:1
```

Example 17. 0600 January 2nd, 2010

```
2010,1.25
```

Example 18. 1200 January 23, 2010

```
2010;023.5000

2010023.5000
```

Example 19. Midnight January 27, 2010

```
2010-027
```

The year may also be omitted in which case it is inferred from the reference point.

A fractional year

This is a number with a fractional component between 1900.0 and 2999.0. The integer part is interpreted as a year and the fractional part is the fraction of the way through the year.

Example 20. Fractional year at 2010-01-01T00:00:00Z

```
2010.0
```

Example 21. Fractional year at 2010-07-02T12:00:00Z

```
2010.5
```

An epoch time

This is a decimal number of seconds since 1900-01-01T00:00:00Z. This may also be begun with `E:` to disambiguate from ISO time strings.

The time specification may be followed by an offset, this consists of a plus or minus, an integer, and a unit. The integer number of the given time unit are added or subtracted to the time specified before. The following time units are valid:

`msec` **or** `milliseconds`

`s`, `sec`, **or** `seconds`

`m`, `min`, **or** `minutes`

Exactly X * 60 seconds.

`h` **or** `hours`

Exactly X * 3600 seconds.

`d` **or** `days`

Exactly X * 86400 seconds.

`w` **or** `weeks`

Exactly X * 604800 seconds.

`mo`, `mon`, **or** `months`

Preserves the time of day and day of month.

`q`, `qtr`, **or** `quarter`

Preserves the time after the start of the quarter.

`y` **or** `years`

Preserves the month, day of month, and time of day.

Example 22. Five day offset

```
2010-03-10T00:15:00Z+5d is equivalent to 2010-03-15T00:15:00Z.
```

NOTE

Overflow

When the alignment would overflow a date field it is set to the largest valid one in the alignment. So one month up from Mar 31 would be on Apr 30.

Bounds List

A bounds specification consists of one or more fields that specify a start and end time. The start time is specified by the leftmost fields. Either bound (start or end) can consist of a single absolute time specification single absolute time specification or an interval offset from the other bound.

The bounds specification can also accept complete specifications of the forms:

A single time

The single time item specifies the start time while the end time is inferred from the logical units of the start.

Example 23. Whole week

```
2003W12
```

Example 24. Whole quarter

```
2001Q2
```

A single integer

The integer can have an optional prefix of `W` or `Q` This selected the whole week or quarter specified by the reference point (by default, the current year). If no unit is given then it is assumed it is a day of year.

Example 25. Week of the current year

```
W12
```

A relative integer

If one bound is available as a complete specification, then the remaining one may be an integer as described above. In this case the year will be inferred from the complete specification.

Example 26. DOY of a selected year

```
2012-05-01 125
```

A four digit year and at least one integer

The first four digit year specifies the year used with the integer specification, which is handled above. Any remaining fields are treated as follows:

No remaining fields

The end is set based on the logical units used used in the specification

A single remaining field

The end time is treated as another integer with the specified start year as the reference point. The time is wrapped to the next year if the end time would be before the start. Two remaining fields with the first a four digit integer: The end year is set from the four digit year. The end time is specified using the same integer selection procedure described above.

Example 27. All of day 231 of 2001

```
2001 321
```

Example 28. All of the second quarter of 2002

```
2002 Q2
```

Example 29. Days 100 to 110 of 2005

```
2005 101 110
```

Two integers with a year between

If two integers as specified above are at the start and end with a four digit year between, then the integers are handled as above, but with the year specifying the end year reference.

Example 30. Weeks 21 to 23 of 2010

```
W21 2010 W23
```

Time Interval or Offset

A time interval is composed of a number followed by an optional unit qualifier and alignment flag. The default units and alignment applied depends on the context the time interval is specified in.

The accepted units of time are:

`msec`, `millisecond`, **or** `milliseconds`

`s`, `sec`, `secs`, `second`, **or** `seconds`

`m`, `min`, `mins`, `minute`, **or** `minutes`

`h`, `hour`, **or** `hours`

`d`, `day`, **or** `days`

`w`, `week`, **or** `weeks`

The first day of the week is Sunday.

`mo`, `mon`, `mons`, `month`, **or** `months`

`q`, `qtr`, `qtrs`, `quarter`, **or** `quarters`

If the first day of the year is one, then the quarters start on days 1, 91, 182, and 274.

`y`, `year`, **or** `years`

No suffix

Default units in effect (often seconds).

Example 31. One hour

```
1h
```

Example 32. Two weeks

```
2 week
```

For milliseconds, seconds, minutes, hours, days and weeks the number can have a fractional component. In this mode the actual interpretation is the largest unit smaller than the specified one that can divide evenly into the specified fraction. For example, specifying 1.5 hours is interpreted as 90 minutes. This becomes important when considering alignment.

A time interval can be thought of as an offset from one time to another that determines the bounds of the interval. This clarification is required to understand the concept of alignment of intervals.

The default alignment depends on the context but it can be explicitly set by adding a suffix to the specification:

`a`, `align`, **or** `aligned`

Enable alignment.

`na`, `noalign`, `noaligned`, `no align`, `no aligned`

Disable alignment.

No suffix

Default alignment (often no alignment).

Example 33. One hour with alignment

```
1ha
```

Example 34. Default units aligned

```
2 align
```

Alignment is relatively easy to understand in the context of a single unit. In this context the alignment just rounds up or down (depending on the context) to the nearest boundary of the logical unit.

Example 35. Simple alignment

If an interval of one hour aligned is specified and applied forward to 2016-01-01T00:15:00Z then the result is 2016-01-01T02:00:00Z.

If it was applied to 2016-01-01T00:00:00Z then the result is 2016-01-01T01:00:00Z.

This concept extends to even the more complex units of time. For example, month alignment applies to the logical month boundaries, even if the number of days per month is not constant.

Alignment becomes more complicated when it is used with a number that is not exactly one. For the variable length and complex intervals (day, week, month, quarter and year) it behaves the same as their simple case, just rounding to the nearest single interval. For the remaining intervals, it will attempt to divide them into the next logical one to get an even interval. In other words, if the interval specified is an integer divisor of the larger one then the alignment takes effect on the outside interval. For the most part, this means intervals like 15 minutes align as expected to the 15 minute intervals after the hour.

Example 36. Mulitple alignment

If an interval of 15 minutes is specified then the alignment boundaries are every 15 minutes in the hour. So if it was applied forward to 2016-01-01T00:13:00Z the result would be 2016-01-01T00:30:00Z. If instead it was applied to 2016-01-01T00:00:00Z the result would be 2016-01-01T00:15:00Z.

If the interval was instead 16 minutes then it would fall back to 1-minute boundaries because 16 doest not divide evenly into 60. If the 16 minute interval was applied to 2016-01-01T00:13:00Z the result would be 2016-01-01T00:29:00Z.