56

Cast Scalar Profile Plot and Data

Device-level and sensor-level cast scalar profile data products are time or cast delimited scalar products for profile data. Cast-delimited options include up casts, down casts, stationary periods, step casts (VPS only), and all casts (up, down, and stationary). Individual casts are grouped daily for plotting purposes only. The cast options, plots and file products are available on cabled profilers (Vertical Profiler System (VPS) in Barkley Canyon and the Buoy Profiler System (BPS) in Saanich Inlet). For other casting / profiling deployments such as from ships or through the ice, only plots are available with the daily option (breaks at midnight PST or 0800 UTC), an example of this case are the Pacific Salmon Foundation (PSF) profiling CTDs, Oxygen Sensors and Fluorometers. This data is acquired in partnership with PSF for their Salish Sea Marine Survival program where community based fishers go out regularly to collect CTD casts at various locations. Here is an example in Data Search: https://data.oceannetworks.ca/DataSearch?location=SSMSP

Please note, the plotting products here are distinct from the Time Series Scalar Profile Plot, as they does not have a time axis. Both sets of plots aim to visualize vertical profile data.

Please consult individual instruments' documentation for more information.

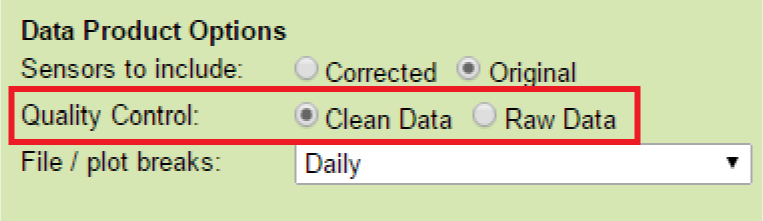
Oceans 2.0 API filter: `dataProductCode=CSPPD`

Revision History

1. 20160901: Initial Release
2. 20180507: Updated with cast delineation, options for cast type selection and temperature/conductivity lag correction

Data Product Options

Quality Control

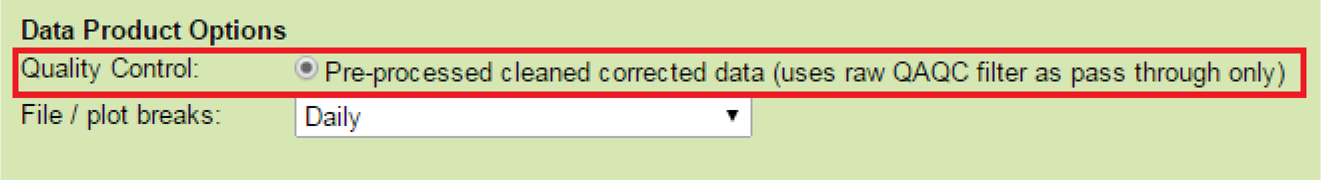


Data Product Options

Sensors to include: ○ Corrected ● Original

Quality Control: ● Clean Data ○ Raw Data

File / plot breaks: Daily ▼



Data Product Options

Quality Control: ● Pre-processed cleaned corrected data (uses raw QAQC filter as pass through only)

File / plot breaks: Daily ▼

Raw Data

When this option is selected, raw data will be supplied in the data products: no action is taken to modify the data. In general, all scalar data is associated with a quality control flag. These flags are stored adjacent to the data values.

Oceans 2.0 API filter: `dpo_qualityControl=0`

Clean Data

Selecting this option will cause any data values with quality control failures (QAQC flags 3, 4 and 6) to be replaced with NaNs. *"-clean"* is added at the end of the filename.

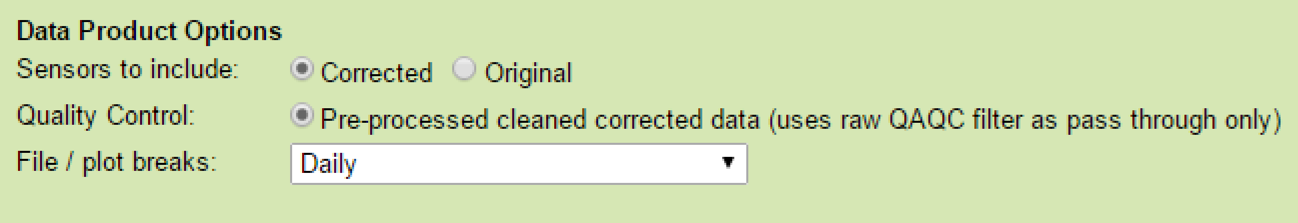
Oceans 2.0 API filter: `dpo_qualityControl=0`

Pre-processed **clean corrected data (uses raw QAQC filter as pass through only)** ***(Only available on corrected profilers and on corrected sensors only)***

There is no quality control option when corrected is chosen for the sensors to include option. The corrected data has already processed and cleaned, so the raw QAQC option is used as a pass-through. Contact us for more information on the processing done to clean and correct cast data from mobile deployments.

Oceans 2.0 API filter: `dpo_qualityControl=0`

Sensors to Include (Device-level Only)


Data Product Options

Sensors to include: ● Corrected ○ Original

Quality Control: ● Pre-processed cleaned corrected data (uses raw QAQC filter as pass through only)

File / plot breaks: Daily ▼

Corrected

A corrected version of the original sensor data that has been re-calculated to factor in variables such as falling speed to improve accuracy. *"-Corrected"* is added at the end of the filename. Sensors without "Corrected" in their names are excluded from the device-level data products.

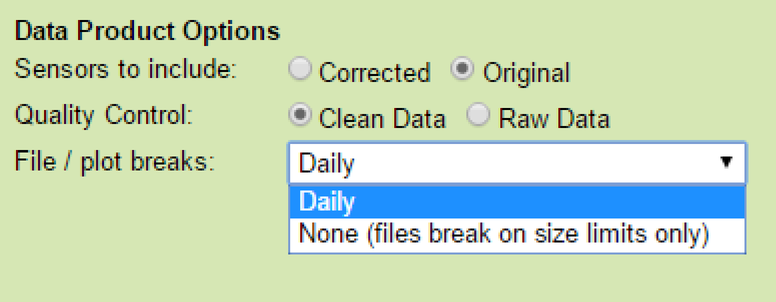
Oceans 2.0 API filter: `dpo_sensorstoinclude=1`

Original

The original sensor data, excludes all sensors with "Corrected" in their name.

Oceans 2.0 API filter: `sensorstoinclude=0`

File / Plot Break


Data Product Options

Sensors to include: ○ Corrected ● Original

Quality Control: ● Clean Data ○ Raw Data

File / plot breaks: Daily ▼

Daily

The time range specified will be broken up daily and a plot will be produced for each day. The break time is not strictly midnight, it is 0800 UTC (midnight PST, 0100 PDT).

Oceans 2.0 API filter: `dpo_fileplotbreaks=1`

None (files break on size limits only)

The time range specified will only be broken up if it exceeds the file size limit (1 GB when loaded into matlab: 400 to 600 MB MAT file).

Oceans 2.0 API filter: `dpo_fileplotbreaks=0`

Cast Delineation / Breaks:

Up casts only

Only data collected while the profiler is ascending the water column are selected. Individual up casts are identified and stored separately within one final data structure for file data products. A plot be produced for each day consisting of all up casts that began during that day. The break time is not strictly midnight, it is 0800 UTC (midnight PST, 0100 PDT).

Oceans 2.0 API filter: `dpo_cast=up_casts`

Down casts only

Only data collected while the profiler is descending the water column are selected. Individual down casts are identified and stored separately within one final data structure. The time range specified will be broken up daily and a plot will be produced for each day consisting of all down casts that began during that day. The break time is not strictly midnight, it is 0800 UTC (midnight PST, 0100 PDT).

Oceans 2.0 API filter: `dpo_cast=down_casts`

Stepped casts only

Only data collected while the profiler is stepped mode are selected (currently unique to VPS). Individual stepped casts (all up casts and stationary segments during a stepped cast) are identified and stored separately within one final data structure. Individual plots will be produced for each stepped cast.

Oceans 2.0 API filter: `dpo_cast=stepped_casts`

All casts

Only data collected while the profiler is ascending, descending, or stationary in the water column are selected. Individual casts (up, down, stationary) are identified and stored separately within one final data structure. The time range specified will be broken up daily and a plot will be produced for each day consisting of casts (up, down, stationary) that began during that day. The break time is not strictly midnight, it is 0800 UTC (midnight PST, 0100 PDT).

Oceans 2.0 API filter: `dpo_cast=all_casts`

Daily

The time range specified will be broken up daily and a plot will be produced for each day. The break time is not strictly midnight, it is 0800 UTC (midnight PST, 0100 PDT).

Oceans 2.0 API filter: `dpo_cast=daily`

None (files break on size limits only)

The time range specified will only be broken up if it exceeds the file size limit (1 GB when loaded into matlab: 400 to 600 MB MAT file).

Oceans 2.0 API filter: `dpo_cast_none`

Formats

PNG / PDF: Cast Scalar Profile Plot

Oceans 2.0 API filter: `extension={png,pdf}`

Cast Scalar Profile plots are plots of depth versus the variable or sensor being plotted and are available as **PNG** or **PDF** format plots. These are the standard CTD cast-type plots. Multiple sensors maybe plotted on multiple x-axes for device-level plots.

For device level plots, particularly for CTDs, we put density, sigma-t, and conductivity vs depth in the first plot, with specific colours while the remaining sensors are plotted in subsequent plots. The second plot contains practical salinity, temperature, sigma-theta, and sound speed vs depth. Each plot is colour coded to improve readability. The second plot will have *"_2"* added on the end of the filename. Here is an example of a device level plot search for down casts on the BPS, resulting in two plots:

Here is an example of single sensor plot for up casts on the VPS:

MAT / netCDF: Cast Scalar Profile Data

This data is available as **MAT** or **netCDF** data products.

MAT Files

Oceans 2.0 API filter: `extension={mat,nc}`

MAT files (v7) can be opened using MathWorks MATLAB 7.0 or later. The file contains two structures: ProfileData and metadata.

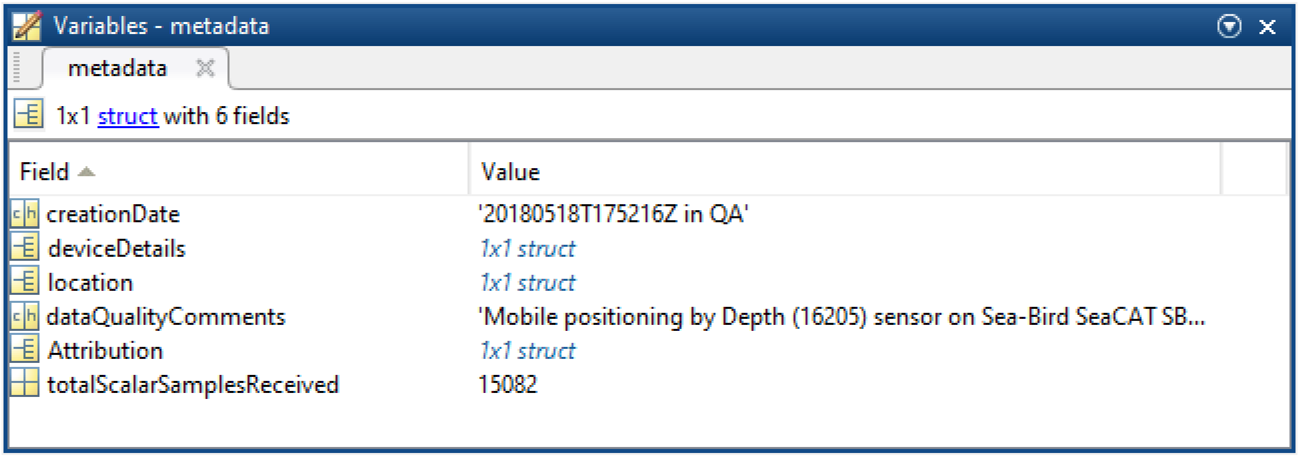
ProfileData: structured as an 1 x N structure with M fields where M includes the sensor data fields (conductivity, temperature, salinity, etc.) and cast details (cast start date, end data, duration, depth, direction, etc.), and N is the cast number:

- Sensor Data (example: seawatertemperature, depth, etc.): see below for structure details
- startDate: Start date of the profile in the format 'dd-mmm-yyyy HH:MM:SS.FFF'
- endDate: End date of the profile in the format 'dd-mmm-yyyy HH:MM:SS.FFF'
- duration: Duration of the profile, in minutes.
- meanDate: Mean date of the profile (mean of startDate and endDate)
- minDepth: Minimum depth of the profile (in meters)
- maxDepth: Maximum depth of the profile (in meters)
- depthRange: Total depth of the profile (in meters)
- direction: Direction of cast (either 'Down Casts', 'Up Casts', or 'Stationary')

Sensor data is stored in a structure identical to its original data structure (the original being the standard scalar data structure defined in Time Series Scalar Data).

- sensorID: Unique identifier number for sensor.
- sensorName: Name of sensor.
- sensorCode: Unique string for the sensor.
- sensorDescription: Description of sensor.
- sensorType: Type of sensor as classified in the ONC data model.
- sensorTypeID: ONC ID given to sensor type.
- units: Unit of measure for the sensor data.
- isEngineeringSensor: boolean (flag) to determine if sensor is an engineering sensor.
- sensorDerivation: String describing the source of the sensor data: derived from calibration formula (dmas-derived), calculated on the device (instrument-derived), calculated by an external process (externally-derived), or direct from the instrument.
- propertyCode: Unique string for the sensor using only lowercase letters and no unique characters
- isMobilePositionSensor: boolean (flag) to determine if sensor is a mobile sensor. Note, this will only be flagged true if this data was added in addition to the requested data. For example, if the user requests a device-level mat product from a GPS device, then the latitude sensor is not flagged. Conversely, if the user requests temperature data from a mobile platform like a ship, then the latitude data from the GPS is added and interpolated to match the time stamps of the temperature sensor. See Positioning and Attitude for Mobile Devices for more information.
- deviceID: Unique identifier number for the parent device.
- searchDateNumFrom: Start date of the specific search in MATLAB datenum format - searches are truncated by availability and deployment dates.
- searchDateNumTo: End date of the specific search MATLAB datenum format - searches are truncated by availability and deployment dates.
- samplePeriod: Vector of sample periods in seconds.
- samplePeriodDateFrom: Vector of the start date of each sample period (MATLAB datenum format).
- samplePeriodDateTo: Vector of the end date of each of sample period (MATLAB datenum format).
- sampleSize: The size of the data sample.
- resampleType: Type of resampling used.
- resampleDescription: Description of the resampleing used.
- resamplePeriod_sec: Resample period in seconds.
- resampleTypeID: Unique identifier of the subsample type used: 0/NaN - none, 1 - average, 2 - decimated (not offered), 3 - min/max, 4 - linear interpolation (VPS pressure only).
- dataProductOptions: A string describing the data product options selected for this data product. This information is reflected in the file name.
- qaqcFlagDescription: A string describing the flags. See the QAQC page for more information.
- time: A vector of data timestamps in MATLAB datenum format.
- dat: A vector of sensor values corresponding to each timestamp. When resampling by averaging, this becomes the average value. (May make a separate field for this in the future, especially if users prefer that option).
- qaqcFlags: A vector indicating the quality of the data, matching the time and dat vectors. See the QAQC page for more information.
- dataDateNumFrom: First time-stamp of the time series.
- dataDateNumTo: Last time-stamp of the time series.
- samplesExpected: The number of valid samples expected from the minimum returned data to the maximum returned data, accounting for variations in sample period.
- samplesReceived: The number of raw samples received, maybe less than length(data.time) when data gaps are being filled with the NaN option.
- Calibration: Structure containing information on the calibration formula applied to the data, as a it appears on the sensor listing page, in the JEP langauage. Fields include: dateFrom, dateTo, sensorID, name, formula.

metadata: a structure array (one structure per device) containing the following metadata fields:

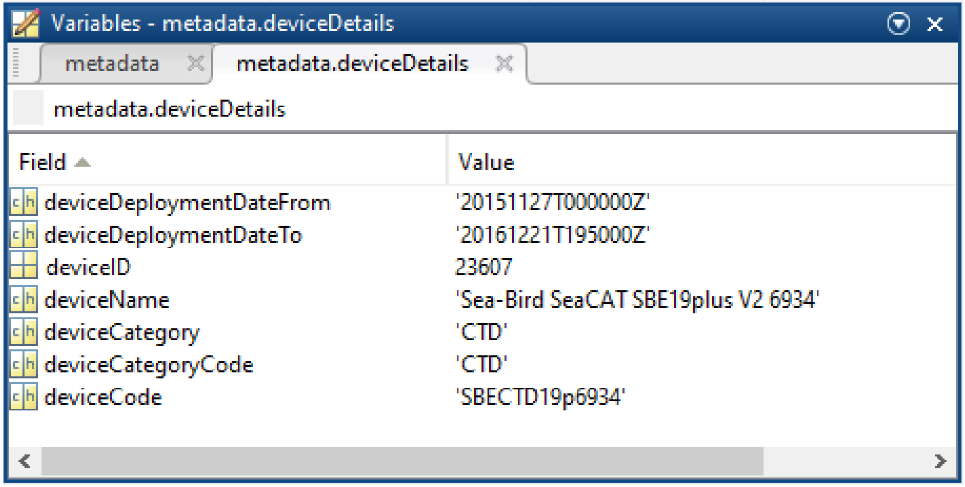


1x1 struct with 6 fields

Field	Value
creationDate	'20180518T175216Z in QA'
deviceDetails	*1x1 struct*
location	*1x1 struct*
dataQualityComments	'Mobile positioning by Depth (16205) sensor on Sea-Bird SeaCAT SB...
Attribution	*1x1 struct*
totalScalarSamplesReceived	15082

- creationDate:Date and time (using ISO8601 format) that the data product was produced. This is a valuable indicator for comparing to other revisions of the same data product.

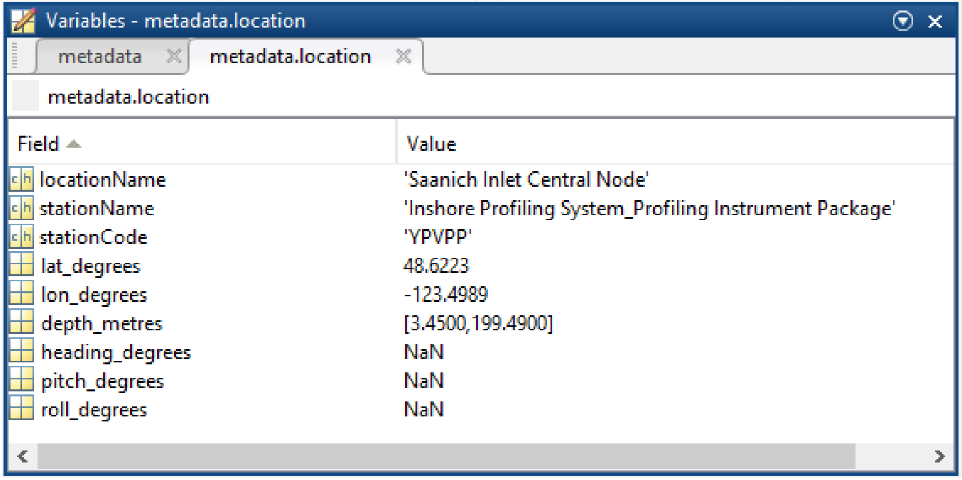
- deviceDetails: a structure array with a structure for each deployment, with the following fields:



Field	Value
deviceDeploymentDateFrom	'20151127T000000Z'
deviceDeploymentDateTo	'20161221T195000Z'
deviceID	23607
deviceName	'Sea-Bird SeaCAT SBE19plus V2 6934'
deviceCategory	'CTD'
deviceCategoryCode	'CTD'
deviceCode	'SBECTD19p6934'

  - deviceDeploymentDateFrom
  - deviceDeploymentDateTo
  - deviceID: A unique identifier to represent the instrument within the ONC observatory.
  - deviceName: A name given to the instrument.
  - deviceCategory: A unique name given to the category of devices, such as 'CTD'
  - deviceCategoryCode: Code representing the device category. Used for accessing webservices, as described here: API / webservice documentation (log in to see this link).
  - deviceCode: A unique string for the instrument which is used to generate instrument search data product file names.

- location: a structure array with a structure for each deployment location, with the following fields:

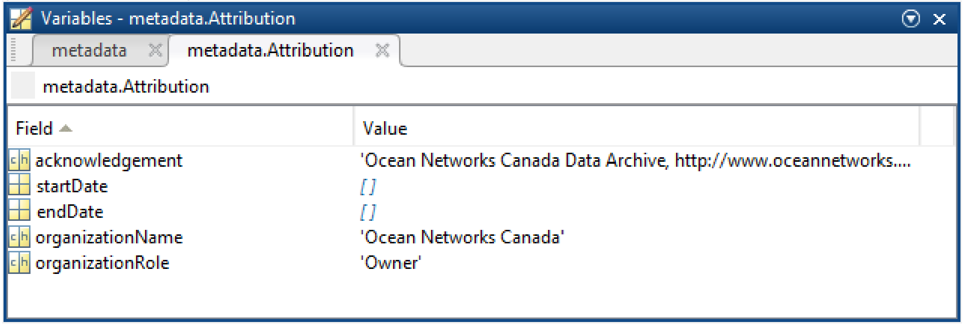


Field	Value
locationName	'Saanich Inlet Central Node'
stationName	'Inshore Profiling System_Profiling Instrument Package'
stationCode	'YPVPP'
lat_degrees	48.6223
lon_degrees	-123.4989
depth_metres	[3.4500,199.4900]
heading_degrees	NaN
pitch_degrees	NaN
roll_degrees	NaN

  - stationName: Secondary location name.
  - stationCode: Code representing the station. Used for accessing webservices, as described here: API / webservice documentation (log in to see this link).
  - depth_metres: Obtained at time of deployment. If NaN, the device is mobile and this position is a variable, the data for which is supplied by a sensor in the data struct.
  - lat_degrees: Obtained at time of deployment. If NaN, the device is mobile and this position is a variable, the data for which is supplied by a sensor in the data struct.
  - lon_degrees: Obtained at time of deployment. If NaN, the device is mobile and this position is a variable, the data for which is supplied by a sensor in the data struct.
  - heading_degrees: Obtained at time of deployment. If NaN, the device is mobile and this position is a variable, the data for which is supplied by a sensor in the data struct.
  - pitch_degrees: Obtained at time of deployment. If NaN, the device is mobile and this position is a variable, the data for which is supplied by a sensor in the data struct.
  - roll_degrees: Obtained at time of deployment. If NaN, the device is mobile and this position is a variable, the data for which is supplied by a sensor in the data struct.

- dataQualityComments: In some cases, there are particular quality-related issues that are mentioned here. This is distinct from QAQC information contained in the data structure.
- Attribution: A structure array with citation information, ordered by importance and date. For internal users, go to the Network Console to configure the citations. If a company has more that one role it will be collated. If there are gaps in the date ranges, they are filled in with the default Ocean Networks Canada citation. If the "Citation Required?" field is set to "No" on the Network Console then the citation will not appear. For data products with a citation (except MAT files) and for users making products from a MAT file, if the special citation is blank/null, then the company default citation will be used and if it is also blank/null, then the final citation will consist of the company name and role: "Ocean Networks Canada

(Owner, Collaborator)". Here are the fields:



Field	Value
acknowledgement	'Ocean Networks Canada Data Archive, http://www.oceannetworks....
startDate	[ ]
endDate	[ ]
organizationName	'Ocean Networks Canada'
organizationRole	'Owner'

  - acknowledgement: the citation text, note that if the special citation is blank/null, the default citation is used.
  - startDate: datenum format
  - endDate: datenum format
  - organizationName
  - organizationRole
- totalScalarSamplesReceived: The number of time stamps that have any valid data on any sensor (at each time stamp). Only defined for metaData (1).totalScalarSamplesReceived, as this total is a summary of all device deployments in the data product.

NETCDF Files

Oceans 2.0 API filter: `extension=nc`

NetCDF is a machine-independent data format offered by numerous institutions, particularly within the earth and ocean science communities. Additional resources are noted here.

Two **NetCDF** files are created with this Data Product: TimeSeriesNetCDF and BinnedNetCDF. The **NetCDF** files are extracted from the data contained in the above **MAT** file.

The **TimeSeriesNetCDF** file contains the following variables:

- time: Time of measurement in days since 1970-01-01 00:00:00
- start_stop_indx: Start and stop indices for each profile (index corresponding to time series)
- direction: Direction of cast (up = -1, down = 1, stationary = 0)
- variable1*: Time series of sensor data (ie seawatertemperature, depth, etc.). The time series only includes data from desired profile (ie only down casts), and the start and stop indices of each profile are stored in the variable start_stop_indx
- variable2*: Time series of next sensor (ie seawatertemperature, depth, etc.).
- variable1_qaqcFlags: Time series of sensor qaqc flag
- variable2_qaqcFlags: Time series of next sensor qaqc flag

* these will be labelled using the sensor's propertyCode (ie depth, depth_qaqcFlags)

Example text file to display contents: InshoreProfilingSystem_ProfilingInstrumentPackage_CTD_Temperature_20160821T000000Z_20160822T000000Z-clean_Profile_DownCasts_TimeSeries.txt

Example NetCDF file: InshoreProfilingSystem_ProfilingInstrumentPackage_CTD_Temperature_20160821T000000Z_20160822T000000Z-clean_Profile_DownCasts_TimeSeries.nc

The **BinnedNetCDF** file contains the following variables:

- time_prof: A single date for each profile (the mean profile date)
- depth_bin: Water depth (in m) of the edges of the measurement bin (ie depth_bin(1) = 0.5 m and depth_bin(2) = 1.5 m so the first bin covers the depths 0.5 to 1.5 m, in other words binned to a 1 m grid centered around integer depths)
- direction: Direction of cast (up = -1, down = 1, stationary = 0)
- variable1*: Binned profile of sensor data (ie seawatertemperature, depth, etc., see DP61 for details on binning method)
- variable2*: Binned profile of next sensor (ie seawatertemperature, depth, etc.)
- variable1*_qaqcFlags: Qaqc flag of each bin for sensor (see “Documentation for Integrating QAQC Flags in VENUS search” for method details)
- variable2*_qaqcFlags: Qaqc flag of each bin for next sensor

* these will be labelled using the sensor's propertyCode (i.e. depth, depth_qaqcFlags)

Unfortunately, our NetCDF files are not CF-Compliant due to the use of ONC variable names, but this will change in the future.

Example text file to display contents: InshoreProfilingSystem_ProfilingInstrumentPackage_CTD_Temperature_20160821T030720Z_20160821T212313Z-clean_Binned.txt

Example NetCDF file: InshoreProfilingSystem_ProfilingInstrumentPackage_CTD_Temperature_20160821T030720Z_20160821T212313Z-clean_Binned.nc

Temperature/Conductivity Lag Correction

A temporal lag between temperature and conductivity sensors can arise from the difference in physical location, response time, and flow rate of the sensors. Another source of T-C lag is due to heat stored in the conductivity sensor material. If users choose this product or Time Series Scalar Profile Plot and Gridded Data and request data that includes temperature and conductivity sensors, they will receive estimated T-C correction terms. These values are used to calculate newly aligned sensors designated with '_aligned' after the sensors name.

T-C lag values are calculated using an FFT process originally written by Jody Klymak. T-C profiles undergo spectral analysis and cases where coherence > 90% are used to calculate a lag correction term that is applied to the temperature sensor, and subsequently applied to the derived sensors (salinity, density, sigma-t, sigma-theta, and sound speed) by re-calculating them with the newly aligned temperature data. These lag correction terms are stored in the data structure under the field 'TClag'. The TClag values are smoothed over 6 casts. If there are less than 6 values to average over, the user will get a warning that 'TClag determined from small sample size'. Lack of any TClag values will result in NaN.

The newly calculated '_aligned' sensors do not have unique sensor ID's. Therefore, they cannot be explicitly searched for in Oceans 2.0 and will not show up in Cast Scalar Profile plots. They will however show up in Time Series Scalar Profile plots and both **MAT** or **netCDF** data files, but only if the user selects data that includes both temperature and conductivity sensors.

Discussion

To comment on this product, click *Write a comment* below.