119

Sound Metrics Sonar Data

Sound Metrics sonars are a non-rotating multibeam sonar, with two main types: the ARIS and Didson sonars. Both are fairly similar: short range very high frequency imaging sonars, producing images like a medical ultrasound (similar technology and centre frequencies). Both ARIS and Didson operate with centre frequencies from 1 to 3 MHz, with 128 beams over a swatch that covers 28 degrees horizontally and 14 degrees vertically, with the beams along the horizontal (in normal deployments). For more information, see the manufacturer's website: http://www.soundmetrics.com. Software for both the ARIS (called *ARIScope*) and Didson can be requested through the Sound Metrics website http://www.soundmetrics.com (go to the support page), see below for more details or contact us.

Revision History

1. April 4th, 2017: Initial release
2. June 7th, 2022: Add support for Didson sonars and new data products

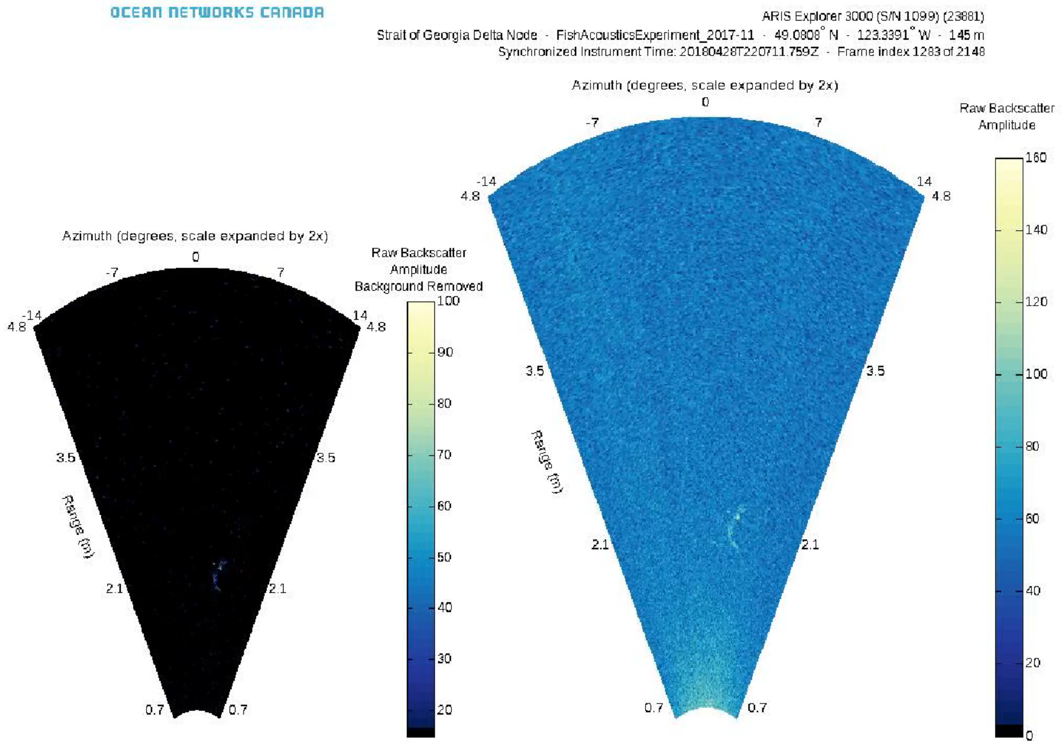
Data Product Options

None at this time.

Formats

The ARIS and Didson sonars output data at such a high rate and density that individual sonar images are not useful. Instead, most users will be interested in the **MP4** videos made from the ARIS (.**aris**) and Didson (.**ddf**) data. The videos use a frame rate equal to that of the live sonar, so the video duration should be the same duration as the duty cycle of the instrument. If the video is shorter than expected, it is likely that a few sonar image frames were dropped due to data transmission congestion (the sonar uses UDP transmission over Ethernet, so there is no guarantee of packet delivery). Raw data is handled by the Oceans 3.0 device driver, archived, then converted to manufacturer's format **ARIS** or **DDF** files. The **DDF** file format is used by DIDSON devices and closely resembles the **ARIS** file format produced and used on the ARIS sonars. Both the **DDF** and **ARIS** file types can be read and analyzed using Sound Metrics software, and is also supported in Echoview https://echoview.com/. **MAT** files, which are generated from **ARIS** or **DDF** files, are available on-demand. **MAT** files are an intermediate product from which users can make plots or do analysis. The Sound Metrics community has MATLAB code for reading both files types - we make use of parts of that code to produce our **MAT** files, so that these files contain the same outputs as the community code and more. **MP4** video files, also generated from **ARIS** or **DDF** files (via the MAT file structure), are archived and stored for fast retrieval, but will be generated on-the-fly if missing (or more likely, delayed).

Each sonar and video frame has an accurate time stamp from the ARIS sonar itself. This time is synchronized by the device driver every hour and included in all the data product formats. The Didson **DDF** files are time stamped with the network time so every time stamp is accurate. This frame time will be slightly different from the time in the file-names as that time is when the data was received at the shorestation, the difference being the normal latency over the network. Here's an example:

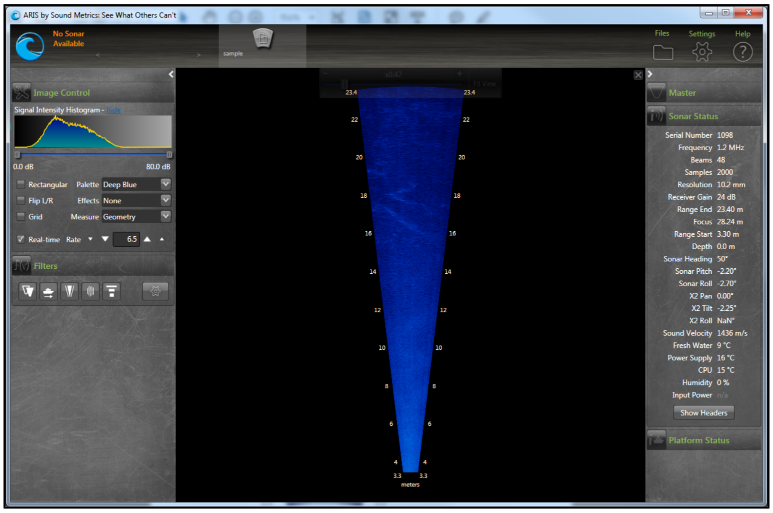


Left: background removed by subtracting the hourly median of each sample to show movement. Right: all data. Movement detected.

Plot generated 03-Nov-2018 04:39:58 UTC

Manufacturer's binary file formats: ARIS and DDF

These files are Sound Metric's custom format files for the ARIS and Didson sonar, used by the *ARIScope* and *Didson* software. Here's an example from *ARIScope*, playing back an **ARIS** file:



□ Rectangular Palette Deep Blue
□ Flip L/R Effects None
□ Grid Measure Geometry
☑ Real-time Rate 6.5

Serial Number 1098
Frequency 1.2 MHz
Beams 48
Samples 2000
Resolution 10.2 mm
Receiver Gain 24 dB
Range End 23.40 m
Focus 28.24 m
Range Start 3.30 m
Depth 0.0 m
Sonar Heading 50°
Sonar Pitch -2.20°
Sonar Roll -2.70°
X2 Pan 0.00°
X2 Tilt -2.25°
X2 Roll NaN°
Sound Velocity 1436 m/s
Fresh Water 9 °C
Power Supply 16 °C
CPU 15 °C
Humidity 0 %
Input Power n/a

The widely used Echoview software offers support and further details of the formats here: https://support.echoview.com/WebHelp/Reference/File_formats/DIDSON_data_files.htm#ARIS For **ARIS** files, this source states: "This file format consists of a master file header followed by one or more frames of data. ARIS parameters and data is stored in the master file header and frame headers. Data are displayed in a multibeam echogram where frames correspond to pings ..." Also on that page is a basic description of **DDF** files: "This file format consists of a master file header followed by one or more frames of data. Both snap shot (master header and a single frame) and movie format files (master header and multiple frames) are supported. Data is displayed in a multibeam echogram where frames correspond to pings. Each 8-bit underlying data sample stores a number from 0 to 255 representing 0 to 90 dB when processed - these can be exported as underlying data. No phase information is available."

More information on the **DDF** file format is available at end of the Didson software user's manual: DIDSON-V5-26-Software-Manual.pdf

Here's the specifics on the **DDF** file format: DIDSON V5.26.26 Data File and Ethernet Structure.pdf

Here's a guide for the Didson: DIDSON_Handbook_Sept2009.pdf

Here's a guide for the ARIS that gives a good overview of its operation: ARIS 3000-X2 Manual.pdf

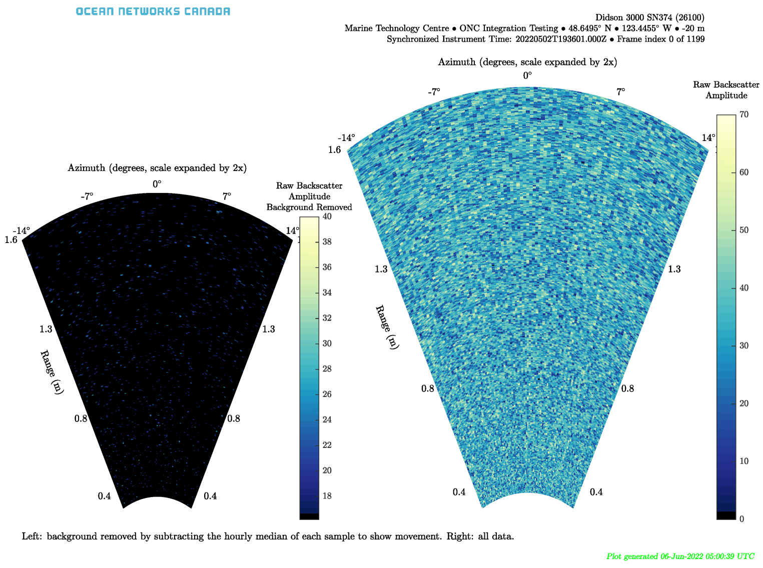
Additional resources (code, more file specifications) are available through Sound Metrics support forum (users need to create account) or contact us.

Imaging formats: MP4, PNG, PDF, GIF

As described above, the **MP4** format is a video of the sonar images / frames, very similar to what a user would see in the *ARIScope* and *Didson* software.

In addition to the **ARIS** and **DDF** manufacturer's binary file format, there are existing **MAT** data file and **MP4** image products. We have recently added new image file formats - **PNG**, **PDF**, and **GIF** . Basic metadata information is included in the plot titles. These are just like the **MP4** format, but are subsets of all the frames and are offered in higher resolution (our normal 3200 by 2400 300 dpi PNGs). Plots are generated for every 60 seconds worth of data as background, and then every 5 seconds when a fish or movement is detected.

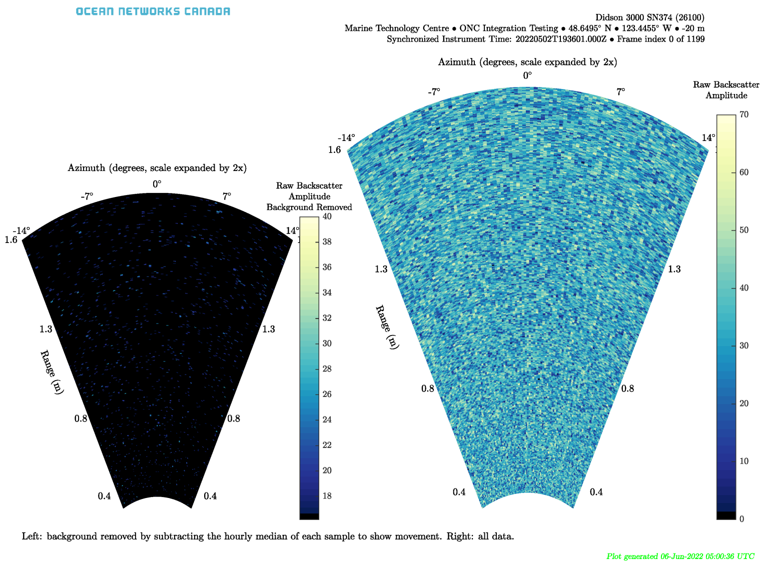
The **GIF** file format:



Left: background removed by subtracting the hourly median of each sample to show movement. Right: all data.

Plot generated 06-Jun-2022 05:00:39 UTC

The **PNG** file format, which includes multiple separate files for the frames:



Left: background removed by subtracting the hourly median of each sample to show movement. Right: all data.

Plot generated 06-Jun-2022 05:00:36 UTC

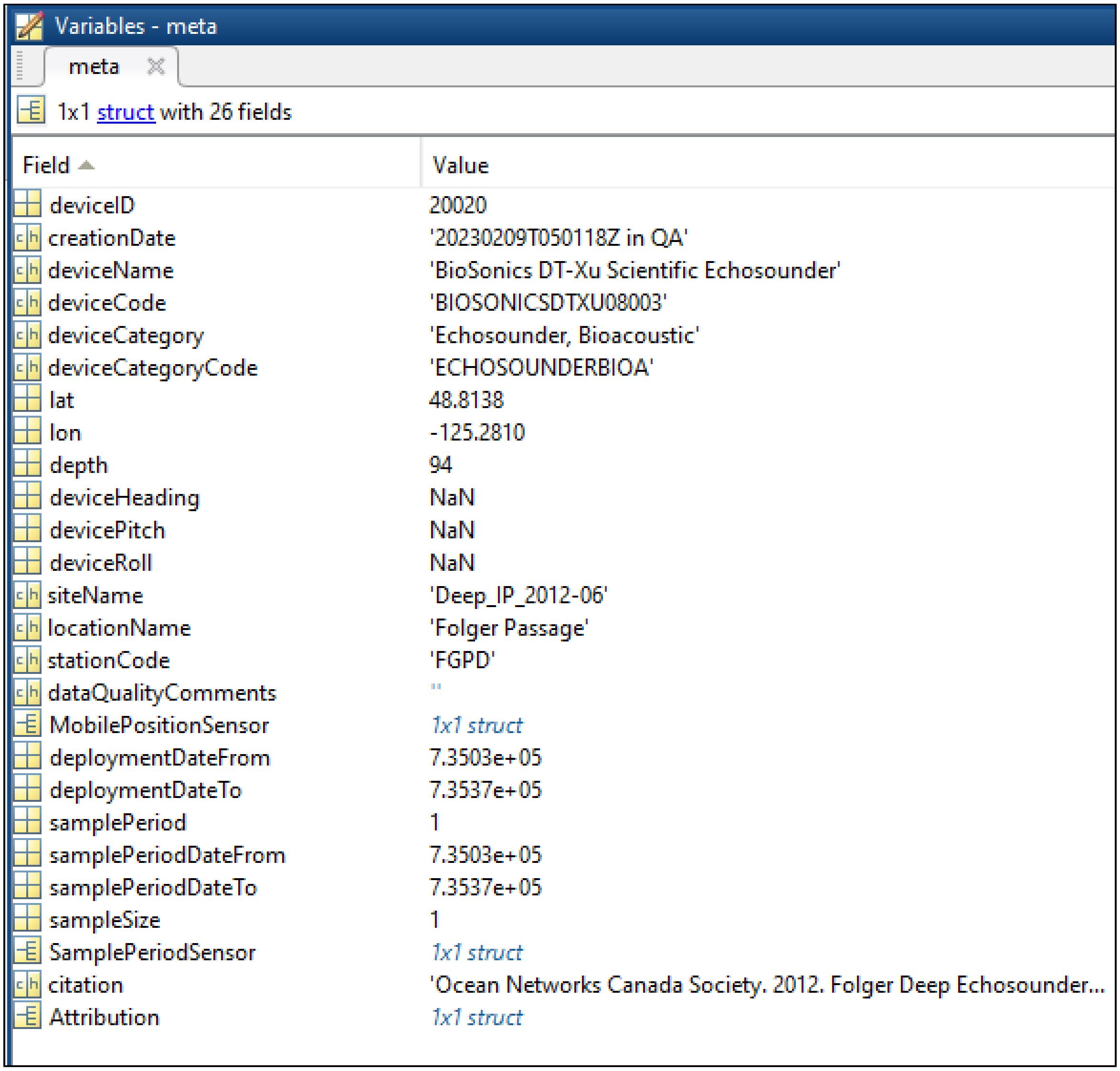
The **PDF** file format, which includes a page for each frame:

DIDSON3000SN3...4100.000Z.pdf

MAT

MAT files (v7) can be opened using Mathworks Matlab 7.0 or later. The file contains three structures, Meta, Data and Config. Many of the fields listed below may not be included in the **MAT** file products as they are from different versions and configurations.

Meta: a structure array containing the following metadata fields:

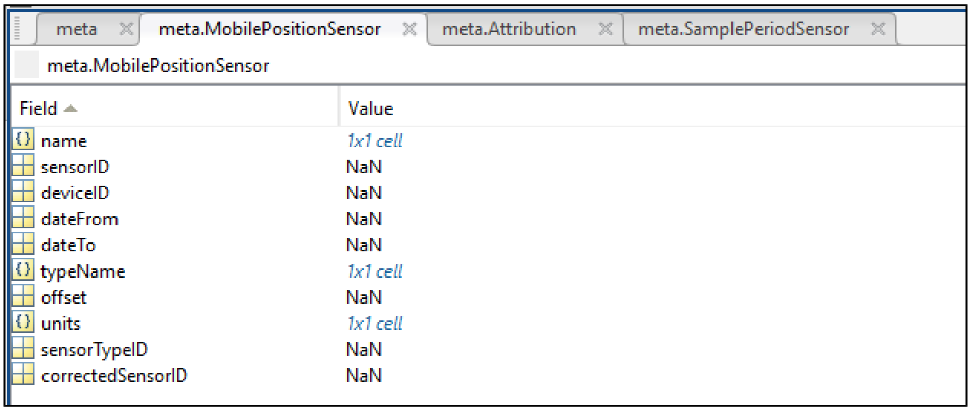


1x1 struct with 26 fields

Field	Value
deviceID	20020
creationDate	'20230209T050118Z in QA'
deviceName	'BioSonics DT-Xu Scientific Echosounder'
deviceCode	'BIOSONICSDTXU08003'
deviceCategory	'Echosounder, Bioacoustic'
deviceCategoryCode	'ECHOSOUNDERBIOA'
lat	48.8138
lon	-125.2810
depth	94
deviceHeading	NaN
devicePitch	NaN
deviceRoll	NaN
siteName	'Deep_IP_2012-06'
locationName	'Folger Passage'
stationCode	'FGPD'
dataQualityComments	''
MobilePositionSensor	*1x1 struct*
deploymentDateFrom	7.3503e+05
deploymentDateTo	7.3537e+05
samplePeriod	1
samplePeriodDateFrom	7.3503e+05
samplePeriodDateTo	7.3537e+05
sampleSize	1
SamplePeriodSensor	*1x1 struct*
citation	'Ocean Networks Canada Society. 2012. Folger Deep Echosounder...
Attribution	*1x1 struct*

- deviceID: A unique identifier to represent the instrument within the Ocean Networks Canada data management and archiving system.
- 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.
- deviceName: A name given to the instrument.
- deviceCode: A unique string for the instrument which is used to generate data product filenames.
- deviceCategory: Device category to list under data search ('Echosounder').
- deviceCategoryCode: Code representing the device category. Used for accessing webservices, as described here: API / webservice documentation (log in to see this link).
- lat: Fixed value obtained at time of deployment. Will be NaN if mobile or if both site latitude and device offset are null. If mobile, sensor information will be available in mobilePositionSensor structure..
- lon: Fixed value obtained at time of deployment. Will be NaN if mobile or if both site longitude and device offset are null. If mobile, sensor information will be available in mobilePositionSensor structure.
- depth: Fixed value obtained at time of deployment. Will be NaN if mobile or if both site depth and device offset are null. If mobile, sensor information will be available in mobilePositionSensor structure.
- deviceHeading: Fixed value obtained at time of deployment. Will be NaN if mobile or if both site heading and device offset are null. If mobile, sensor information will be available in mobilePositionSensor structure.
- devicePitch: Fixed value obtained at time of deployment. Will be NaN if mobile or if both site pitch and device offset are null. If mobile, sensor information will be available in mobilePositionSensor structure.
- deviceRoll: Fixed value obtained at time of deployment. Will be NaN if mobile or if both site roll and device offset are null. If mobile, sensor information will be available in mobilePositionSensor structure.
- siteName: Name corresponding to its latitude, longitude, depth position.
- locationName: The node of the Ocean Networks Canada observatory. Each location contains many sites.

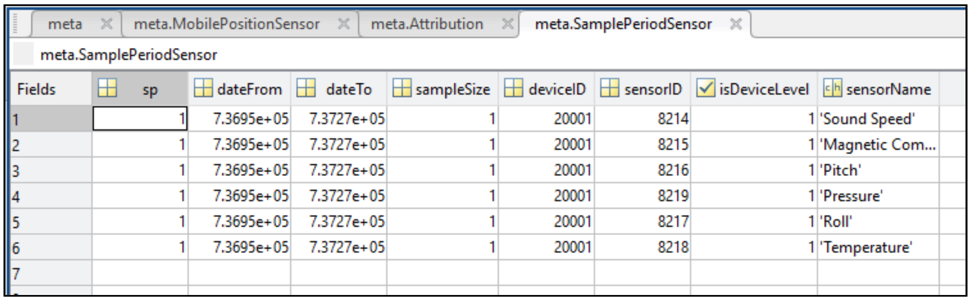
- stationCode: Code representing the station or site. Used for accessing webservices, as described here: API / webservice documentation (log in to see this link).
- dataQualityComments: In some cases, there are particular quality-related issues that are mentioned here.
- MobilePositionSensor: A structure with information about sensors that provide additional scalar data on positioning and attitude (latitude, longitidue, depth below sea surface, heading, pitch, yaw, etc).



meta.MobilePositionSensor

Field	Value
name	*1x1 cell*
sensorID	NaN
deviceID	NaN
dateFrom	NaN
dateTo	NaN
typeName	*1x1 cell*
offset	NaN
units	*1x1 cell*
sensorTypeID	NaN
correctedSensorID	NaN

  - name: A cell array of sensor names for mobile position sensors. If not a mobile device, this will be an empty cell string.
  - sensorID: An array of unique identifiers of sensors that provide position data for mobile devices - this data may be used in this data product.
  - deviceID: An array of unique identifiers of devices that provide position data for mobile devices - this data may be used in this data product.
  - dateFrom: An array of datenums denoting the range of applicability of each mobile position sensor - this data may be used in this data product.
  - dateTo: An array of datenums denoting the range of applicability of each mobile position sensor - this data may be used in this data product.
  - typeName: A cell array of sensor names for mobile position sensors. If not a mobile device, this will be an empty cell string. One of: Latitude, Longitude, Depth, COMPASS_SENSOR, Pitch, Roll.
  - offset: An array of offsets between the mobile position sensors' values and the position of the device (for instance, if cabled profiler has a depth sensor that is 1.2 m above the device, the offset will be -1.2m).
  - sensorTypeID: An array of unique identifiers for the sensor type.
  - correctedSensorID: An array of unique identifiers of sensors that provide corrected mobile positioning data. This is generally used for profiling deployments where the latency is corrected for: CTD casts primarily.
- deploymentDateFrom: The date of the deployment on which the data was acquired.
- deploymentDateTo: The date of the end of the deployment on which the data was acquired (will be NaN if still deployed).
- samplingPeriod: Sample period / data rating of the device in seconds, this is the sample period that controls the polling or reporting rate of the device (some parsed scalar sensors may report faster, some devices report in bursts) (may be omitted for some data products).
- samplingPeriodDateFrom: matlab datenum of the start of the corresponding sample period (may be omitted for some data products).
- samplingPeriodDateTo: matlab datenum of the end of the corresponding sample period (may be omitted for some data products).
- sampleSize: the number of readings per sample period, normally 1, except for instruments that report in bursts. Will be zero for intermittent devices (may be omitted for some data products).
- SamplePeriodSensor: A structure array with an entry for each scalar sensor on the device (even though this metadata is for complex data products that don't use scalar sensors).

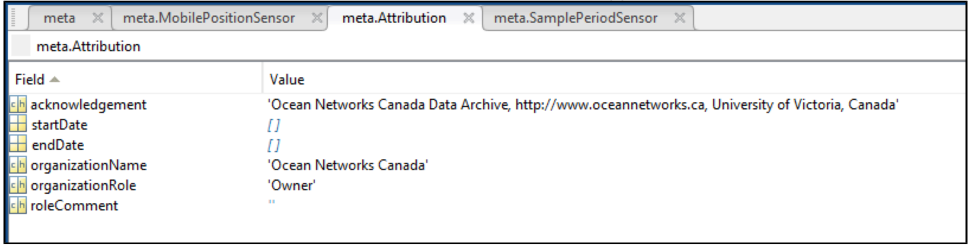


meta.SamplePeriodSensor

Fields	sp	dateFrom	dateTo	sampleSize	deviceID	sensorID	isDeviceLevel	sensorName
1	1	7.3695e+05	7.3727e+05	1	20001	8214	1	'Sound Speed'
2	1	7.3695e+05	7.3727e+05	1	20001	8215	1	'Magnetic Com...
3	1	7.3695e+05	7.3727e+05	1	20001	8216	1	'Pitch'
4	1	7.3695e+05	7.3727e+05	1	20001	8219	1	'Pressure'
5	1	7.3695e+05	7.3727e+05	1	20001	8217	1	'Roll'
6	1	7.3695e+05	7.3727e+05	1	20001	8218	1	'Temperature'
7								

  - sp: sample period in seconds (array), unless sensorid is NaN then this is the device sample period
  - dateFrom: array of date from / start date (inclusive) for each sample period in MATLAB datenum format.
  - dateTo: array of date to / end date (exclusive) for each sample period in MATLAB datenum format.
  - sampleSize: the number of readings per sample period (array). Normally 1, except for instruments that report in bursts. Will be zero for intermittent devices.
  - deviceID: array of unique identifiers of devices (should all be the same).
  - sensorID: array of unique identifiers of sensors on this device.
  - isDeviceLevel: flag (logical) that indicates, when true or 1, if the corresponding sample period/size is from the device-level information (i.e. applies to all sensors and the device driver's poll rate).
  - sensorName: the name of the sensor for which the sample period/size applies (much more user friendly than a sensorID).
- citation: a char array containing the DOI citation text as it appears on the Dataset Landing Page. The citation text is formatted as follows: <Author(s) in alphabetical order>. <Publication Year>. <Title, consisting of Location Name (from searchTreeNodeName or siteName in ONC database) Deployed <Deployment Date (sitedevicedatefrom in ONC database)>. <Repository>. <Persistent Identifier, which is either a DOI URL or the queryPID (search_dtlid in ONC database)>. Accessed Date <query creation date (search.datecreated in ONC database)>
- Attribution: A structure array with information on any contributors, ordered by importance and date. If an organization has more than 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 "Attribution Required?"

field is set to "No" on the Network Console then the citation will not appear. Here are the fields:



meta.Attribution

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

- acknowledgement: the acknowledgement text, usually formatted as "<organizationName> (<organizationRole>)", except for when there are no attributions and the default is used (as shown above).
- startDate: datenum format
- endDate: datenum format
- organizationName
- organizationRole: comma separated list of roles
- roleComment: primarily for internal use, usually used to reference relevant parts of the data agreement (may not appear)

Data: a structure with the following fields:

- frameNumber – frame number in file
- frameTime - PC time stamp when recorded; seconds since epoch (Jan 1st 1970). Units in seconds. For DDF files, this time is always accurate. For ARIS files, the times are corrected in the data product derived from the ARIS files, but in the ARIS files themselves, this time may not be accurate - likely to drift during a deployment, the driver may update the internal time periodically.
- frameTimeStr - corrected frame time in ISO format (yyyymmddTHHMMSSS.FFFZ)
- frameTimeDateNum - corrected frame time in MATLAB's datenum format (number of days since year zero).
- sonarTimeStamp - on-sonar microseconds since epoch (Jan 1st 1970)
- Year - sonar CPU year
- Month - 1-12
- Day - 1-31
- Hour - 0-23
- Minute - 0-59
- Second - 0-59
- HSecond - 0-99
- humidity - relative humidity
- Battery - volts at sonar PS in tenths, e.g. 145 = 14.5 Volts
- velocity - platform velocity from AUV integration
- depth - platform depth from AUV integration
- altitude - not documented
- pitch - platform pitch from AUV integration
- pitchRate - platform pitch rate from AUV integration
- roll - platform roll from AUV integration
- rollRate - platform roll rate from AUV integration
- heading - platform heading from AUV integration
- headingRate - platform heading rate from AUV integration
- compassHeading - sonar compass heading output
- compassPitch - sonar compass pitch output
- compassRoll - sonar compass roll output
- beamTilt - not documented
- targetRange - for topside software use
- targetBearing - for topside software use
- targetPresent - for topside software use
- flags – Bad Data points in the acoustic data
- WaterTemp - water temperature from housing temperature sensor
- timerPeriod - for non-integral frame rates
- sonarX - sonar X location for 3D processing
- sonarY - sonar Y location for 3D processing
- sonarZ - sonar Z location for 3D processing
- sonarPan – X2 pan output
- sonarTilt – X2 Tilt output
- sonar Roll– X2 Roll output
- panPNNL - legacy custom data from Battelle PNNL equipment
- tiltPNNL - legacy custom data from Battelle PNNL equipment
- rollPNNL - legacy custom data from Battelle PNNL equipment
- vehicleTime - special for Bluefin HAUV or other AUV integration
- SonarPanOffset - sonar mount location pan offset for 3D processing; meters
- SonartiltOffset - sonar mount location tilt offset for 3D processing; meters
- SonarRollOffset - sonar mount location roll offset for 3D processing; meters
- SonarXOffset - sonar mount location X offset for 3D processing; meters
- SonarYOffset - sonar mount location Y offset for 3D processing; meters
- SonarZOffset - sonar mount location Z offset for 3D processing; meters
- tmatrix - 3D processing transformation 4x4 matrix
- accellX - X-axis sonar acceleration
- accellY - Y-axis sonar acceleration
- accellZ- Z-axis sonar acceleration
- arisErrorFlagsUint - error flag code bits
- missedPackets - missed packet count for Ethernet statistics reporting
- salinity - water salinity code: 0 = fresh, 15 = brackish, 35 = salt

- pressure - depth sensor output, units in psi
- batteryVoltage - battery input voltage before power steering, units in mV
- mainVoltage - main cable input voltage before power steering, units in mV
- switchVoltage - input voltage after power steering; filtered voltage, units in mV
- focusMotorMoving - for AutomaticRecording
- voltageChanging - 00 = not changing, 01 = turning on, 10 = turning off
- focusTimeoutFault - not documented
- focusOverCurrentFault - not documented
- focusNotFoundFault - not documented
- focusStalledFault - not documented
- fPGATimeoutFault - not documented
- fPGABusyFault - not documented
- fPGAStuckFault - not documented
- cPUTempFault - not documented
- pSUTempFault - not documented
- waterTempFault - not documented
- humidityFault - not documented
- pressureFault - not documented
- voltageReadFault - not documented
- voltageWriteFault - not documented
- focusCurrentPosition - focus shaft current position, in motor units 0..1000
- targetPan - commanded pan position
- targetTilt - commanded tilt position
- targetRoll - commanded roll position
- panMotorErrorCode - not documented
- toltMotorErrorCode - not documented
- rollMotorErrorCode - not documented
- panAbsPosition - low-resolution magnetic encoder absolute pan position
- tiltAbsPosition - low-resolution magnetic encoder absolute tilt position
- rollAbsPosition - low-resolution magnetic encoder absolute roll position
- panAccelX - accelerometer outputs from AR2 CPU board sensor, units of G
- panAccelY - accelerometer outputs from AR2 CPU board sensor, units of G
- panAccelZ - accelerometer outputs from AR2 CPU board sensor, units of G
- tiltAccelX - accelerometer outputs from AR2 CPU board sensor, units of G
- tiltAccelY - accelerometer outputs from AR2 CPU board sensor, units of G
- tiltAccelZ - accelerometer outputs from AR2 CPU board sensor, units of G
- rollAccelX - accelerometer outputs from AR2 CPU board sensor, units of G
- rollAccelY - accelerometer outputs from AR2 CPU board sensor, units of G
- rollAccelZ - accelerometer outputs from AR2 CPU board sensor, units of G
- invalidSettings - cookie indices for command acknowledge in frame header
- upTime - total time the sonar has been running over its lifetime, units in seconds
- goTime -sonar time when frame cycle is initiated in hardware
- panVelocity - AR2 tilt velocity, units in degrees/second
- tiltVelocity - AR2 tilt velocity, units in degrees/second
- rollVelocity - AR2 tilt velocity, units in degrees/second
- acousticData – sound pressure amplitude of return, having dimensions of Config.numBeams by Config.samplesPerBeam by Config.numFrames, datatype is usually uint8

Config: a structure with the following fields:

- version - ARIS file format version
- status - feedback on client commands
- numFrames - number of frames in the file
- frameRate - pings or frames taken per second
- resolution - not documented
- numBeams - number of beams (usually 128)
- sampleRate - calculated as 1e6/SamplePeriod (along beam sample rate for digitalization, probably of the demodulated amplitude envelope, seems to be in Hz)
- fileWindowsStart - file header value for windowStart (length in meters from the transducer to the start of the recording window), this is set value at start up, may not be what is actually used
- fileWindowLength - file header value for windowLength (length in meters from the transducer to the end of the recording window), this is set value at start up, may not be what is actually used
- pingMode - ARIS ping mode, 1..12
- frequencyHiLow - 1 = HF, 0 = LF 4..100 microseconds
- pulseWidth - width of transmit pulse,
- cyclePeriod - ping cycle time, 1802...65535 microseconds
- samplePeriod - downrange sample rate, 4...100 microseconds
- transmitEnable - 1 = transmit ON, 0 = transmit OFF
- frameRate – frames per second
- soundSpeed - sound velocity in water calculated from water temperature depth and salinity setting, units of m/s
- samplesPerBeam - number of downrange samples in each beam (one value per ping)
- enable150V - 1 = 150V on (max power), 0 = 150V off (min power, 12V)
- sampleStartDelay - delay from transmit until start of sampling (window start) in usec, [930..65535]
- largeLens - 1 = telephoto lens (large lens, big lens, hi-res lens) present
- theSystemType - 1 = ARIS 3000, 0 = ARIS 1800, 2 = ARIS 1200
- sonarSerialNumber - sonar serial number as labeled on housing
- arisAppVersion - version number of ArisApp sending frame data
- appliedSettings - cookie indices for command acknowledge in frame header
- constrainedSettings - cookie indices for command acknowledge in frame header
- enableInterpacketDelay - if true delay is added between sending out image data packets

- interpacketDelayPeriod - packet delay factor in us (does not include function overhead time)
- arisAppVersionMajor - major version number
- arisAppVersionMinor - minor version number
- focus - focus units 0-1000
- degC2 - power supply temperature, units in Celsius
- degC1 - CPU temperature, units in Celsius
- receiverGain - 0-40 dB
- intensity -the upper intensity cut off
- threshold – the lower intensity cut off
- windowLength - window length in meters
- windowStart – window start in meters
- latitude - from auxiliary GPS sensor
- longitude - from auxiliary GPS sensor
- sonarPosition - special for PNNL
- configFlags - not documented
- highResolution - non-zero if HF, zero if LF
- numRawBeams - ARIS 3000 = 128/64, ARIS 1800 = 96/48, ARIS 1200 = 48
- samplesPerChannel - Number of range samples in each beam (single value)
- reverse - non-zero = lens down (DIDSON) or lens up (ARIS), zero = opposite
- serialNumber - sonar serial number
- date - date that file was recorded
- idString - user input to identify file in 256 characters
- id1 - user-defined integer quantity
- id2 - user-defined integer quantity
- id3 - user-defined integer quantity
- id4 - user-defined integer quantity
- startFrame - first frame number from source file (for DIDSON snippet files)
- endFrame - last frame number from source file (for DIDSON snippet files)
- timelapse - non-zero indicates time lapse recording
- recordInterval - number of frames/seconds between recorded frames
- radioSeconds - frames or seconds interval
- frameInterval - record every Nth frame
- fileuserAssigned - not documented
- length - not documented
- auxFlags - see DDF_04 file format document
- sspd - sound velocity in water
- flags3D - see DDF_04 file format document %%%fix this
- softwareVersion - DIDSON/ARIS software version that recorded the file
- thumbnailFI - frame index of frame used for thumbnail image of file
- reorderedSamples - 1 = frame data already ordered into [beam,sample] array, 0 = needs reordering
- status1 - display status string on NTSC video systems
- status2 - display status string on NTSC video systems
- panwcom - not documented
- tiltwcom - not documented
- userAssigned - not documented
- samplesPerBeam - number of samples along each beam (can be have a value for each frame/ping)

For those interested in making their own plots from the **MAT** file, the along beam range per sample may be calculated like this:

```
rangeToBinStart = linspace(Config.windowStart(2), Config.windowLength(2) + Config.windowStart(2), size(Data.
acousticData,2));
```

While the angle between beams is a linear spacing with the 28 degree overall "fan-width":

```
theta = ANGLEAMP*linspace(-14*(2*pi)/360,14*2*pi/360,size(Data.acousticData,1));
```

The `ANGLEAMP` is just 2 – it stretches the angles so that it looks better in the **MP4** video.