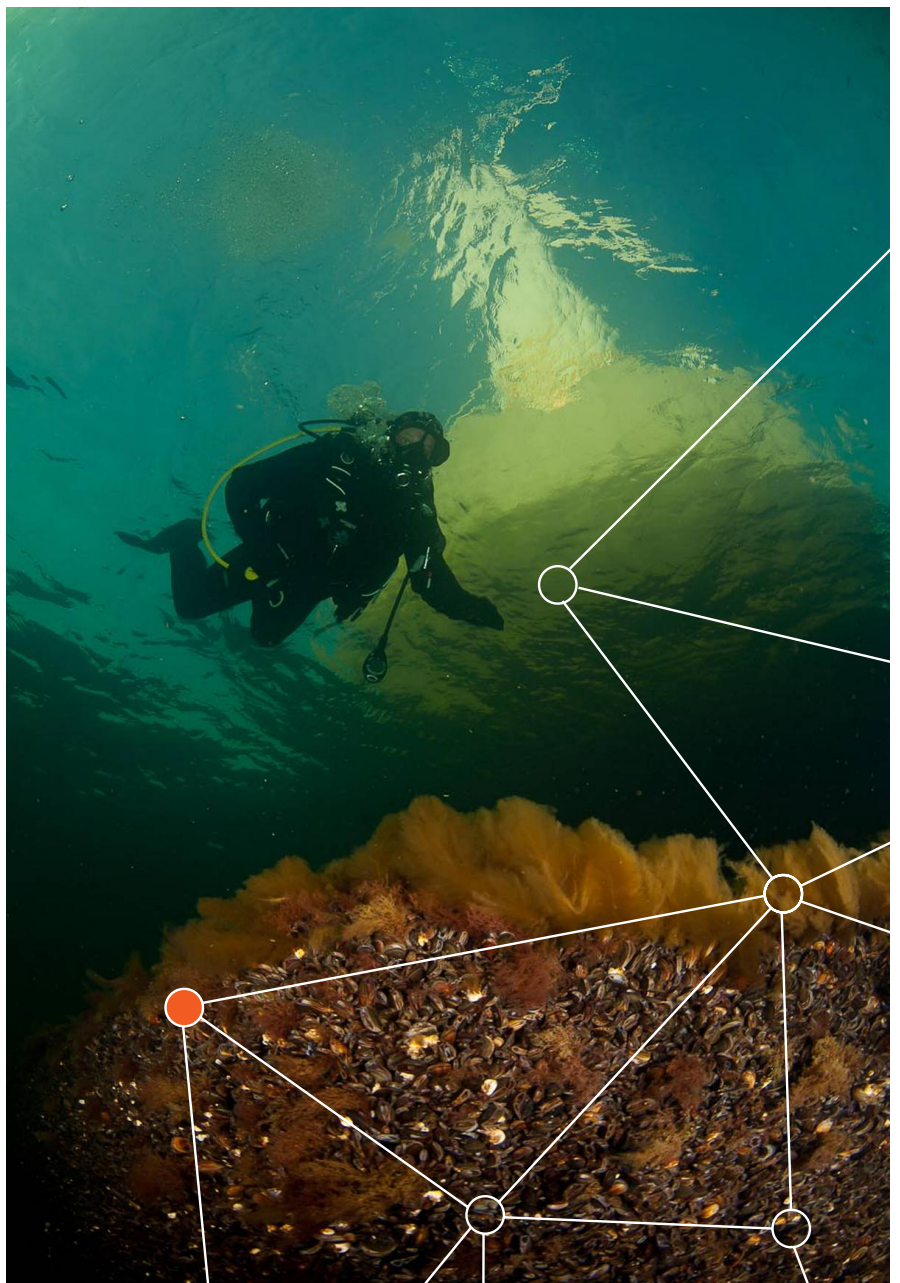


The SONAR-netCDF4 convention for sonar data, Version 1.0

**ICES COOPERATIVE
RESEARCH REPORT**

RAPPORT
DES RECHERCHES
COLLECTIVES



ICES COOPERATIVE RESEARCH REPORT

RAPPORT DES RECHERCHES COLLECTIVES

No. 341

MAY 2018

The SONAR-netCDF4 convention for sonar data, Version 1.0

Editors

Gavin Macaulay • Héctor Peña



ICES

International Council for
the Exploration of the Sea

CIEM

Conseil International pour
l'Exploration de la Mer

International Council for the Exploration of the Sea Conseil International pour l'Exploration de la Mer

H. C. Andersens Boulevard 44–46
DK-1553 Copenhagen V
Denmark
Telephone (+45) 33 38 67 00
Telefax (+45) 33 93 42 15
www.ices.dk
info@ices.dk

Recommended format for purposes of citation:

Macaulay, G. and Peña, H. (Eds.). 2018. The SONAR-netCDF4 convention for sonar data, Version 1.0. ICES Cooperative Research Report No. 341. 33 pp.
<https://doi.org/10.17895/ices.pub.4392>

Series Editor: Emory D. Anderson

The material in this report may be reused for non-commercial purposes using the recommended citation. ICES may only grant usage rights of information, data, images, graphs, etc. of which it has ownership. For other third-party material cited in this report, you must contact the original copyright holder for permission. For citation of datasets or use of data to be included in other databases, please refer to the latest ICES data policy on ICES website. All extracts must be acknowledged. For other reproduction requests please contact the General Secretary.

This document is the product of an expert group under the auspices of the International Council for the Exploration of the Sea and does not necessarily represent the view of the Council.

Cover image: © OCEANA/Carlos Suárez
<https://doi.org/10.17895/ices.pub.4392>
ISBN 978-87-7482-214-1
ISSN 2707-7144

© 2018 International Council for the Exploration of the Sea

Contents

Foreword	i
1 Introduction	1
1.1 Background	1
1.2 Versioning.....	2
2 The SONAR-netCDF4 convention	3
2.1 Introduction.....	3
2.2 Hierarchical structure	3
2.3 Obligations and missing data	4
2.4 Metadata and authorities.....	4
2.5 Units	4
2.6 Datatypes	4
2.7 Vocabularies	4
2.8 File-naming convention	5
2.9 CDL version of SONAR-netCDF4	5
2.10 Groups.....	5
2.10.1 Top-level group	5
2.10.2 Annotation group.....	6
2.10.3 Environment group.....	6
2.10.4 Platform group.....	7
2.10.5 Provenance group	13
2.10.6 Sonar group.....	14
2.10.7 Vendor specific group.....	21
3 Conversion equations	22
3.1 Type 1	22
3.2 Type 2.....	22
4 Coordinate systems	24
4.1 Platform coordinate system.....	24
4.2 Sonar beam coordinate system	25
4.3 Coordinate systems offsets and rotations	25
5 Revision history	26
6 Acknowledgements	27
7 References	28
Annex 1: Working with netCDF4 files	29
Annex 2: Contact information	32

Foreword

This report documents a convention for the storage of sonar data in netCDF4-formatted computer files, with an initial focus on omnisonars. The intention is to provide a well-founded convention that is supported by multiple sonar systems and multiple sonar analysis software packages, with the aim of facilitating the use of sonar data for research and survey purposes. The name of this convention is SONAR-netCDF4.

This document was developed by the Topic Group on Defining a data format for omnidirectional fisheries sonar, part of ICES Working Group on Fisheries Acoustics, Science, and Technology (WGFAST).

1 Introduction

Sonars have long been used to study and understand fisheries and the aquatic environment, but only recently have they directly provided digital data for quantitative analysis. Each manufacturer typically provides such data in a proprietary, but usually open, file format specific to their sonar systems. This hinders the effective use and exchange of such data by requiring the development and maintenance of file-reading software for multiple analysis programs and multiple sonars.

This document presents a convention for the storage and exchange of fisheries sonar data, with an initial focus on omnisonars. It is sufficiently generic and flexible to contain all foreseeable types of fisheries sonar data, along with necessary metadata. It also serves as a statement of the minimum set of data and metadata required to use omnisonar data in a quantitative manner.

1.1 Background

Many purpose-built file formats exist to store and exchange data from scientific and industrial equipment. Formats have been created by sonar and echosounder manufacturers; in addition, more generic acoustic data formats such as the Generic Water Column (GWC) format (Gee *et al.*, 2012), the eXtended Triton Format (XTF; Triton, 2013), and the HydroACoustics (HAC) format (McQuinn *et al.*, 2005) have also been created. These formats store a time-ordered sequence of datagrams, making it easy to append new data. Other data, such as geographical position, are typically interleaved into the sequence of datagrams. However, this is not optimal for analysis purposes when data are viewed and analysed as a set of pings and metadata from a time-period or spatial grouping. In addition, efficient random-read access to individual pings is not possible unless an index is available or created.

The Topic Group on Defining a data format for omnidirectional fisheries sonar do not want to create yet another sonar-specific file format; the knowledge required is not within the expertise of this group. However, the group does have the expertise to specify what data must be stored to allow for unambiguous use of backscatter data during quantitative analysis.

Accordingly, an existing file-format definition has been utilized and what should be stored is specified. The requirements for such a file format definition were:

- ability to adequately represent the content and structure of sonar data and associated metadata;
- standardized, open-file format;
- fast random access to data stored in the file;
- ability to store multiple types of data (e.g. position and backscatter);
- ability to store metadata (e.g. sonar settings);
- freely available and open libraries to read the data files into programming languages and technical computing environments (e.g. Java, C, C++, Python, Matlab, and R);
- reliable and space-efficient format for data exchange and storage;
- self-describing file format;
- backwards compatibility upon modification of the file contents specification (i.e. old software/tools can still read relevant parts of a newer version);
- computer platform-independent;
- long-term support and extensive use in other scientific fields;
- support for very large datasets.

Other scientific communities that collect and produce voluminous amounts of data have addressed similar needs, resulting in the Hierarchical Data Format (The HDF Group, 2017), currently at version 5 (HDF5). This is the only file format that meets all the listed requirements and is utilized for sonar data. There are two realizations of the HDF5 file format: (i) HDF5 itself and (ii) netCDF4 (Unidata, 2017), which is a subset of HDF5. Both are sufficient, but netCDF4 is more widely used and has slightly wider language support and implementation diversity. Accordingly, netCDF4 has been chosen.

Using a well-supported file format has the significant benefit that many data exploration, query, extraction, and analysis tools already read such files. This eliminates the need to develop and maintain sonar-specific file-reading software and facilitates the use of existing tools for data management, distribution, and analysis (a pertinent example is OPeNDAP, which provides a network-transparent standard for distributing scientific data).

An attractive feature of netCDF4 is the ease and transparency with which data can be added to an existing netCDF4 file. For example, processed data and how it was obtained can be included in the same file as the raw data. In addition, netCDF4 files can be very large while still providing fast access to data subsets. This allows a single SONAR-netCDF4 file to contain data from a transect or an entire survey. These features would simplify data management, improve traceability and long-term storage of analysis, enhance the sharing of processed datasets, and facilitate analysis of large disparate datasets. In general, any amount of additional data can be stored in the files without affecting the ability to use the data specified by the SONAR-netCDF4 convention.

A distinction is commonly made between file formats designed for archiving data in the original form, formats designed for storing partially or fully processed data, and formats for data exchange (Jackson *et al.*, 2014). The SONAR-netCDF4 convention is intended to be suitable for all of these.

The initial focus of this convention has been to specify the storage of backscatter amplitude data from omnisonars. It is envisaged that future versions could specify how to store data from other acoustic equipment (e.g. echosounders) and derived data (e.g. bottom and school detections, categorization of backscatter, and integrated backscatter at multiple resolutions). Ancillary information such as sonar display screen-grabs could also be stored.

1.2 Versioning

This document and the convention that it describes will change over time to implement enhancements and to correct errors and omissions. To accommodate this, the SONAR-netCDF4 convention always requires a version number. This document has a separate version number, allowing for revised versions of the document to describe the same version of the convention.

The convention version number will always be included in the title of the document. The document version number will always be found in Section 5.

2 The SONAR-netCDF4 convention

2.1 Introduction

NetCDF4 is a data model, application programming interface (API) library, and file format for storing and managing data, developed and maintained by Unidata. Unidata is part of the US University Corporation for Atmospheric Research (UCAR) Community Programs (UCP) and funded primarily by the US National Science Foundation.

SONAR-netCDF4 is a data and metadata convention for the storage of data from active sonars in netCDF4 formatted files. SONAR-netCDF4 consists primarily of a naming convention and a data structure within the netCDF4 data model.

Datasets can be added to SONAR-netCDF4 files if they do not conflict with the SONAR-netCDF4 datasets. If such additions are potentially of use to other users of the file format, it is recommended that they be proposed for inclusion in this or additional convention specifications.

Each SONAR-netCDF4 file is intended to store data from one sonar mounted on one platform. The storage of data from multiple sonars and multiple platforms in one SONAR-netCDF4 file is not in the scope of the convention.

A design principle of SONAR-netCDF4 has been to focus on describing the acoustic backscatter data, not the overall purpose and context of why the data were collected. Such broader metadata are better stored in separate metadata systems and schema (e.g. ISO, 2014; ICES, 2016).

2.2 Hierarchical structure

NetCDF4 has two main organizational concepts: (i) the variable, which can contain a variety of data structures, and (ii) groups, being a collection of variables. Groups and variables can have attributes attached to them. Groups can be arranged into a hierarchy.

SONAR-netCDF4 divides sonar data into seven netCDF4 groups:

1. **Top-level** – contains metadata about the SONAR-netCDF4 file format;
2. **Annotation** – contains time-stamped annotations;
3. **Environment** – contains information relevant to acoustic propagation through water;
4. **Platform** – contains information about the platform on which the sonar is installed;
5. **Provenance** – contains metadata about how the SONAR-netCDF4 version of the data were obtained;
6. **Sonar** – contains the backscatter and associated metadata; groups under Sonar are used for storing data from different sonar operating modes;
7. **Vendor specific** – contains vendor-specific information about the sonar and the data.

These groups contain variables and variable attributes with prescribed names and contents.

2.3 Obligations and missing data

Some variables and attributes in SONAR-netCDF4 are mandatory; these form the minimal set of data required to quantitatively use backscattering amplitude data. The remaining variables and attributes have various levels of optionality and provide enhanced context and information about the sonar data. The obligations are:

M	mandatory
MA	mandatory if applicable or available
R	recommended
O	optional

Any non-mandatory variables can be absent from a SONAR-netCDF4 file. If a variable is mandatory, it must be present and must contain data. The set of mandatory variables and attributes has been chosen so that omnisonar systems can directly generate SONAR-netCDF4-conforming files without needing survey, experiment, or cruise-specific data.

The `_FillValue` attribute should be used to indicate missing data in variables. For floating point values, the IEEE 754 not-a-number (NaN) is the preferred fill value as this is convenient for commonly-used analysis packages (e.g. Python, Matlab, R).

2.4 Metadata and authorities

The fisheries acoustics community has developed a metadata convention for processed acoustic data (ICES, 2016). Where relevant, attribute and variable names have been reused from this convention. The NetCDF Climate and Forecast (CF) Metadata Conventions (Eaton *et al.*, 2017) have been used where sensible (the efficient storage of unprocessed active sonar data has not been a design goal of the CF convention) along with the ESIP Attribute Convention for Data Discovery (ACDD). Terms and concepts from other metadata conventions have also been used where appropriate.

2.5 Units

All relevant variables and attributes in SONAR-netCDF4 files are required to have a textual netCDF4 attribute with the name "units" that specifies the units. The International System of Units (SI) is used. For simplicity, the data format mandates the use of particular units and their textual form, as per the definitions and conventions of the UDUNITS-2 package (UCAR, 2014). Decibels are commonly used in underwater acoustic quantities, but UDUNITS does not currently include such a unit. However, as the CF convention does permit decibel (with symbols: "dB", "db", and "dbel"), it is used in the SONAR-netCDF4 convention.

2.6 Datatypes

Each item has a suggested datatype, chosen to have sufficient range and precision to contain the expected data. Alternative datatypes can be used if necessary, but are discouraged. The "string" type should contain text in the UTF-8 encoding and should be treated as case-sensitive during any comparisons. Enumerated datatypes are used for some of the controlled vocabularies.

2.7 Vocabularies

The contents of some variables and attributes are restricted to defined vocabularies. These are listed or referenced where required. Desired additions to the vocabularies

should be proposed to WGFAST for incorporation into this document. Some of the vocabularies have been represented as netCDF4 enumeration data types and some using the CF flag_values convention.

2.8 File-naming convention

SONAR-netCDF4 files should always end with a “.nc” suffix to indicate that they are a netCDF file. It is recommended that the filename should sort alphanumerically into chronological order (e.g. date and time of the first ping in the file; thus: YYYYMMDD-HHMMSS.nc). This facilitates file management and use in analysis systems.

2.9 CDL version of SONAR-netCDF4

An example of a SONAR-netCDF4 file is provided in the Common Data form Language (CDL). This provides a more formalized and structured representation of the data format. A CDL file can be converted into a NetCDF file using the “ncgen” utility (available as part of the netCDF software distribution) and then used as a template for creating populated SONAR-netCDF4 files.

2.10 Groups

2.10.1 Top-level group

The top-level group contains metadata about the SONAR-netCDF4 file, represented as attributes in the group (Table 1).

Table 1. Description of the top-level group.

Description	Obligation	Comment
Group attributes		
:Conventions = "CF-1.7, SONAR-netCDF4-1.0, ACDD-1.3"	M	A comma-separated list of the conventions followed in the file. Include the SONAR-netCDF4 convention and version (e.g. "SONAR-netCDF4-1.0") and the relevant CF and ACDD conventions (e.g. "CF-1.7" and "ACDD-1.3").
:date_created	M	Timestamp of file creation in ISO8601:2004 extended format, including the time zone (e.g. 2017-05-06T20:21:35Z).
:keywords	M	A comma-separated list of key words and/or phrases. For direct sonar-generated files, this should at least include the type of sonar.
:license	O	Either enter the URL to a standard or specific license, enter "Freely distributed" or "None", or describe any restrictions to data access and distribution in free text.
:rights	O	Description of the usage rights of data in the file.
:sonar_convention_authority = "ICES"	M	Name of the organization managing and distributing the SONAR-netCDF4 convention. Currently ICES.

:sonar_convention_name = "SONAR-netCDF4"	M	Formal name of this convention (i.e. "SONAR-netCDF4").
:sonar_convention_version = "1.0"	M	SONAR-netCDF4 version number in the form "major.minor", where major and minor are non-negative integers.
:summary	M	A paragraph describing the dataset, analogous to an abstract for a paper. For direct sonar-generated files, this can be blank.
:title	M	A short phrase or sentence describing the dataset. For direct sonar-generated files, this can be as simple as "Files generated by the XYZ sonar".

2.10.2 Annotation group

The annotation group contains timestamped annotations with optional identification code. Annotations are typically textual notes entered by users relevant to the data at a particular time. Some sonar systems provide an interface for creating manual and programmatic annotations. The netCDF4 group name is **/Annotation** and is described in Table 2.

Table 2. Description of the annotation group.

Description	Obligation	Comment
Dimensions		
time = unlimited		Can be of fixed or unlimited length, as appropriate.
Coordinate variable		
uint64 time(time)	MA	
:axis = "T"		
:calendar = "gregorian"		
:long_name = "Timestamps of annotations"		
:standard_name = "time"		
:units = "nanoseconds since 1601-01-01 00:00:00Z"		
Variables		
string annotation_category(time)	O	Optional category for the annotation, for use in grouping annotation types.
:long_name = "Annotation category"		
string annotation_text(time)	MA	
:long_name = "Annotation text"		

2.10.3 Environment group

The environment group contains information on environmental conditions, especially the speed of sound in water and acoustic absorption. The netCDF4 group name is

/Environment and is described in Table 3. Sound speed, absorption, and current profiles can also be stored in this group, as can profile measurements of salinity, temperature, and pressure. Such profile data should use the NCEI NetCDF “profile” template, v2.0 or greater.

Table 3. Description of the environment group.

Description	Obligation	Comment
Dimensions		
frequency = unlimited		Can be of fixed or unlimited length, as appropriate.
Coordinate variables		
float frequency(frequency)	M	
:long_name = "Acoustic frequency"		
:standard_name = "sound_frequency"		
:units = "Hz"		
float :valid_min = 0.0		
Variables		
float absorption_indicative(frequency)	M	Indicative absorption values used to calculate the time-varied-gain (TVG), in the absence of more detailed data.
:long_name = "Indicative acoustic absorption"		
:units = "dB/m"		
float :valid_min = 0.0		
float sound_speed_indicative	M	Mean sound speed in water used to calculate echo range, in the absence of more detailed sound-speed data.
:long_name = "Indicative sound speed"		
:standard_name = "speed_of_sound_in_sea_water"		
:units = "m/s"		
float :valid_min = 0.0		

2.10.4 Platform group

This group contains information about the sonar platform (e.g. ship). The netCDF4 group name is **/Platform** and is described in Table 4. The coordinate system convention, including the definition of pitch and roll, is detailed in Section 4.

Optionally, **/Platform** subgroups can be included to store data from individual instruments. This would provide a standard place to store unprocessed and unparsed instrument data from multiple GPS sensors, motion reference units, etc. The data format is not prescribed, but could, for example, be NMEA-style text and/or numeric values. Subgroups must have one attribute called “description” that provides a short description of the contents. Other attributes can be added as desired. The variables under the subgroup should have appropriate names and an attribute that gives the units, where

appropriate. If NMEA telegrams are stored, it is suggested that the group name is NMEA and follows the convention described in Table 5.

Table 4. Description of the platform group.

Description	Obligation	Comment
Group attributes		
:platform_code_ICES	O	Platform code from the ICES SHIPC vocabulary (http://vocab.ices.dk/services/pox/GetCodeList/SHIPC).
:platform_name	O	Platform name of which the sonar is a part.
:platform_type	O	Platform type that the sonar is part of. Use the description field from the ICES "Platform Class" vocabulary (http://vocab.ices.dk/services/pox/GetCodeList/Platform%20Class).
Dimensions		
time1 = unlimited		Can be of fixed or unlimited length, as appropriate.
time2 = unlimited		Can be of fixed or unlimited length, as appropriate. Use as many time-dimensions as necessary with names of timeX, where X is a positive integer.
Coordinate variables		
uint64 time1(time1)		Use as many time-coordinate variables as necessary to contain the sample times of the various platform variable sensors. Use the naming scheme of timeX, where X is a positive integer.
:axis = "T"		
:calendar = "gregorian"		
:long_name = "Timestamps for position data"		
:standard_name = "time"		
:units = "nanoseconds since 1601-01-01 00:00:00Z"		
uint64 time2(time2)		Use as many time-coordinate variables as necessary to contain the sample times of the various platform variable sensors. Use the naming scheme of timeX, where X is a positive integer.
:axis = "T"		
:calendar = "gregorian"		
:long_name = "Timestamps for gyrocompass data"		
:standard_name = "time"		
:units = "nanoseconds since 1601-01-01 00:00:00Z"		
Variables		
float distance(time2)	O	Distance travelled by the platform from an arbitrary location.
:long_name = "Distance travelled by the platform"		
:units = "m"		

:valid_min = "0.0"		
float heading(time2)	MA	Platform heading. Measured clockwise from north.
:long_name = "Platform heading (true)"		
:standard_name = "platform_orientation"		
:units = "degrees_north"		
float :valid_range = 0.0, 360.0		
double latitude(time1)	MA	Platform latitudes in the WGS-84 coordinate reference system.
:long_name = "Platform latitude"		
:standard_name = "latitude"		
:units = "degrees_north"		
double :valid_range = -90.0, 90.0		
double longitude(time1)	MA	Platform longitudes in the WGS-84 coordinate reference system.
:long_name = "Platform longitude"		
:standard_name = "longitude"		
:units = "degrees_east"		
double :valid_range = -180.0, 180.0		
float MRU_offset_x	R	<i>x</i> -axis component of the vector from the platform coordinate system origin to the motion reference unit origin.
:long_name = "Distance along the <i>x</i> -axis from the platform coordinate system origin to the motion reference unit sensor origin"		
:units = "m"		
float MRU_offset_y	R	<i>y</i> -axis component of the vector from the platform coordinate system origin to the motion reference unit origin.
:long_name = "Distance along the <i>y</i> -axis from the platform coordinate system origin to the motion reference unit sensor origin"		
:units = "m"		
float MRU_offset_z	R	<i>z</i> -axis component of the vector from the platform coordinate system origin to the motion reference unit origin.
:long_name = "Distance along the <i>z</i> -axis from the platform coordinate system origin to the motion reference unit sensor origin"		

:units = "m"		
float MRU_rotation_x	R	Extrinsic angular rotation about the x -axis from the platform zero angle to the MRU zero angle.
:long_name = "Extrinsic rotation about the x -axis from the platform to MRU coordinate systems"		
:units = "arc_degree"		
float :valid_range = -180.0, 180.0		
float MRU_rotation_y	R	Extrinsic angular rotation about the y -axis from the platform zero angle to the MRU zero angle.
:long_name = "Extrinsic rotation about the y -axis from the platform to MRU coordinate systems"		
:units = "arc_degree"		
float :valid_range = -180.0, 180.0		
float MRU_rotation_z	R	Extrinsic angular rotation about the z -axis from the platform zero angle to the MRU zero angle.
:long_name = "Extrinsic rotation about the z -axis from the platform to MRU coordinate systems"		
:units = "arc_degree"		
float :valid_range = -180.0, 180.0		
float pitch(time2)	MA	Platform pitch. Positive values indicate a bow-up pitch.
:long_name = "Platform pitch"		
:standard_name = "platform_pitch_angle"		
:units = "arc_degree"		
float :valid_range = -90.0, 90.0		
float position_offset_x	R	Distance from the platform coordinate system origin to the latitude/longitude position origin along the x -axis.
:long_name = "Distance along the x -axis from the platform coordinate system origin to the latitude/longitude sensor origin"		
:units = "m"		
float position_offset_y	R	Distance from the platform coordinate system origin to the latitude/longitude position origin along the y -axis.

:long_name = "Distance along the <i>y</i> -axis from the platform coordinate system origin to the latitude/longitude sensor origin"		
:units = "m"		
float position_offset_z	R	Distance from the platform coordinate system origin to the latitude/longitude position origin along the <i>z</i> -axis.
:long_name = "Distance along the <i>z</i> -axis from the platform coordinate system origin to the latitude/longitude sensor origin"		
:units = "m"		
float roll(time2)	MA	Platform roll. Positive values indicate a roll to starboard.
:long_name = "Platform roll"		
:standard_name = "platform_roll_angle"		
:units = "arc_degree"		
float :valid_range = -180.0, 180.0		
float speed_ground(time1)	MA	Platform speed over ground.
:long_name = "Platform speed over ground"		
:standard_name = "platform_speed_wrt_ground"		
:units = "m/s"		
float :valid_min = 0.0		
float speed_relative(time2)	O	Platform speed relative to water.
:long_name = "Platform speed relative to water"		
:standard_name = "platform_speed_wrt_seawater"		
:units = "m/s"		
float :valid_min = 0.0		
float transducer_offset_x	R	Distance from the platform coordinate system origin to the monostatic sonar transducer along the <i>x</i> -axis.
:long_name = "x-axis distance from the platform coordinate system origin to the sonar transducer"		
:units = "m"		
float transducer_offset_y	R	Distance from the platform coordinate system origin to the monostatic sonar transducer along the <i>y</i> -axis.

:long_name = "y-axis distance from the platform coordinate system origin to the sonar transducer"		
:units = "m"		
float transducer_offset_z	R	Distance from the platform coordinate system origin to the monostatic sonar transducer along the z-axis.
:long_name = "z-axis distance from the platform coordinate system origin to the sonar transducer"		
:units = "m"		
float vertical_offset(time2)	R	Distance from the nominal water level to the actual water level measured along the z-axis of the platform coordinate system (positive values are when the actual water level is below the nominal water level). For ships and similar, this is called heave, but the concept applies equally well to underwater vehicle depth. This offset is applied at the position given by (MRU_offset_x, MRU_offset_y, MRU_offset_z).
:long_name = "Platform vertical offset from nominal"		
:units = "m"		
float water_level	R	Distance from the origin of the platform coordinate system to the nominal water level measured along the z-axis of the platform coordinate system (positive values are below the origin). The distance between the nominal and actual water level is provided by vertical_offset.
:long_name = "Distance from the platform coordinate system origin to the nominal water level along the z-axis"		
:units = "m"		
Subgroups		
NMEA	O	Suggested subgroup to store raw NMEA data as an example of a /Platform subgroup.

Table 5. Suggested group for storing NMEA datagrams from marine instruments.

Description	Obligation	Comment
Group attributes		
:description = "All NMEA sensor datagrams"	M	Description of the subgroup contents.
Dimensions		
time = unlimited		Can be of fixed or unlimited length, as appropriate.
Coordinate variables		
uint64 time(time)	M	
:axis = "T"		
:calendar = "gregorian"		
:long_name = "Timestamps for NMEA datagrams"		
:standard_name = "time"		
:units = "nanoseconds since 1601-01-01 00:00:00Z"		
Variables		
string NMEA_datagram(time)	O	Example of how to store NMEA datagrams.
:long_name = "NMEA datagram"		

2.10.5 Provenance group

The provenance group provides information on how the SONAR-netCDF4 version of the data were created. The netCDF4 group name is **/Provenance** and is described in Table 6.

Table 6. Description of the provenance group.

Description	Obligation	Comment
Group attributes		
:conversion_software_name	MA	Name of the software used to do the conversion.
:conversion_software_version	MA	Version of the software used to do the conversion.
:conversion_time	MA	Date and time of the start of the conversion process in extended ISO8601:2005 extended format, including time zone.
Dimensions		
filenames = unlimited		Can be of fixed or unlimited length, as appropriate.
Variables		
string source_filenames(filenames)	MA	Vector of datafile names that were used to generate the data in this SONAR-netCDF4 file.
:long_name = "Source filenames"		

2.10.6 Sonar group

This group contains the sonar backscatter data and associated metadata. The netCDF4 group name is **/Sonar** and is described in Table 7.

Data from each beam mode (e.g. horizontal and vertical beam modes) are stored in subgroups under the **/Sonar** group (see Table 8). The form of the backscatter data can vary between different sonar systems. For example, some provide a complex-valued amplitude, while others provide a real- or integer-valued amplitude. Variable definitions for data from split-aperture systems are not currently specified.

Table 7. Description of the sonar group.

Description	Obligation	Comment
Group attributes		
:sonar_manufacturer	R	Name of the sonar manufacturer.
:sonar_model	R	Name of the sonar model.
:sonar_serial_number	R	Sonar serial number.
:sonar_software_name	R	Sonar software name.
:sonar_software_version	R	Sonar software version.
:sonar_type = "omni-sonar"	M	Type of sonar, chosen from a defined vocabulary (currently only one value): "omnisonar" (a sonar with a nominally omnidirectional mode).
Types		
byte enum beam_stabilisation_t {not_stabilised = 0, stabilised = 1}		Whether or not the beam direction is compensated for platform motion.
byte enum beam_t {single = 0, split_aperture = 1}		Beam type. Split aperture indicates a beam that can detect the arrival angle of echoes, while single beam cannot.
byte enum conversion_equation_t {type_1 = 1, type_2 = 2}		The type of equation used to convert backscatter measurements into volume backscattering and target strength.
byte enum transmit_t {CW = 0, LFM = 1, HFM = 2}		Type of transmit pulse. CW = continuous wave – a pulse nominally of one frequency, LFM = linear frequency modulation – a pulse which varies from transmit_frequency_start to transmit_frequency_stop in a linear manner over the nominal pulse duration (transmit_duration_nominal), HFM = hyperbolic frequency modulation – a pulse which varies from transmit_frequency_start to transmit_frequency_stop in a hyperbolic manner over the nominal pulse duration (transmit_duration_nominal).
float(*) sample_t		Variable length vector used to store ragged arrays of backscatter data.
Subgroups		
Beam_group1	M	Example of a beam group. Include as many subgroups as necessary for different beam groups. Use unique group names, preferably of the form Beam_groupX where X is an integer.

Table 8. Description of the beam mode subgroups of the sonar group.

Description	Obligation	Comment
Group attributes		
:beam_mode	M	Mode of the beam in this subgroup, taken from the defined vocabulary of: "vertical" (a set of beams that form a vertical slice through the water), "horizontal" (a set of beams that form a nominally horizontal plane through the water), and "inspection" (a set of beams with arbitrary pointing directions).
conversion_equation_t :conversion_equation_type	M	Type of equation used to convert backscatter measurements into volume backscattering strength and target strength.
Dimensions		
beam		The number of beams in this beam group.
ping_time = unlimited		Can be of fixed or unlimited length, as appropriate.
Coordinate variables		
string beam(beam)	M	Beam name (or number or identification code).
:long_name = "Beam name"		
uint64 ping_time(ping_time)	M	Timestamp at which each ping occurred.
:axis = "T"		
:calendar = "gregorian"		
:long_name = "Time-stamp of each ping"		
:standard_name = "time"		
:units = "nanoseconds since 1601-01-01 00:00:00Z"		
Variables		
sample_t backscatter_i(ping_time, beam)	MA	Imaginary part of backscatter measurements. Each element in the 2D matrix is a variable length vector (of type sample_t) that contains the samples for that beam and ping time.
:long_name = "Raw backscatter measurements (imaginary part)"		
:units = "as appropriate"		Use units appropriate for the data.

sample_t backscatter_r(ping_time, beam)	M	Real part or amplitude or power of backscatter measurements. Each element in the 2D matrix is a variable length vector (of type sample_t) that contains the samples for that beam and ping time.
:long_name = "Raw backscatter measurements (real part)"		
:units = "as appropriate"		Use units appropriate for the data.
float beamwidth_receive_major (ping_time, beam)	M	One-way beam width at half power down in the horizontal direction of the receive beam.
:long_name = "Half power one-way receive beam width along major (horizontal) axis of beam"		
:units = "arc_degree"		
float :valid_range = 0.0, 360.0		
float beamwidth_receive_minor (ping_time, beam)	M	One-way beam width at half power down in the vertical direction of the receive beam.
:long_name = "Half power one-way receive beam width along minor (vertical) axis of beam"		
:units = "arc_degree"		
float :valid_range = 0.0, 360.0		
float beamwidth_transmit_major (ping_time, beam)	MA	One-way beam width at half power down in the horizontal direction of the transmit beam.
:long_name = "Half power one-way transmit beam width along major (horizontal) axis of beam"		
:units = "arc_degree"		
float :valid_range = 0.0, 360.0		
float beamwidth_transmit_minor (ping_time, beam)	MA	One-way beam width at half power down in the vertical direction of the transmit beam.
:long_name = "Half power one-way transmit beam width along minor (vertical) axis of beam"		
:units = "arc_degree"		
float :valid_range = 0.0, 360.0		
float beam_direction_x(ping_time, beam)	M	The x-axis coordinate of a unit vector in the beam direction for each beam and ping, as per the sonar beam coordinate system.

:long_name = "x-component of the vector that gives the pointing direction of the beam, in sonar beam coordinate system"		
:units = "1"		
float :valid_range = -1.0, 1.0		
float beam_direction_y(ping_time, beam)	M	The <i>y</i> -axis coordinate of a unit vector in the beam direction for each beam and ping, as per the sonar beam coordinate system.
:long_name = " <i>y</i> -component of the vector that gives the pointing direction of the beam, in sonar beam coordinate system"		
:units = "1"		
float :valid_range = -1.0, 1.0		
float beam_direction_z(ping_time, beam)	M	The <i>z</i> -axis coordinate of a unit vector in the beam direction for each beam and ping, as per the sonar beam coordinate system.
:long_name = " <i>z</i> -component of the vector that gives the pointing direction of the beam, in sonar beam coordinate system"		
:units = "1"		
float :valid_range = -1.0, 1.0		
beam_stabilisation_t beam_stabilisation(ping_time)	M	Indicates whether or not sonar beams have been compensated for platform motion.
:long_name = "Beam stabilisation applied (or not)"		
beam_t beam_type(ping_time)	M	Type of beam.
:long_name = "Type of beam"		
float equivalent_beam_angle(ping_time, beam)	M	Equivalent beam angle.
:long_name = "Equivalent beam angle"		
:units = "sr"		
float :valid_range = 0.0, 12.56637061435917295385		Maximum value is equivalent to 4π .
float gain_correction(ping_time, beam)	MA	Gain correction. This parameter is set from a calibration exercise. Necessary for type 2 conversion equation.

:long_name = "Gain correction"		
:units = "dB"		
short non_quantitative_processing (ping_time)	M	Settings of any processing that is applied prior to recording backscatter data that may prevent the calculation of calibrated backscatter. A value of 0 always indicates no such processing.
:flag_meanings		Space-separated list of non-quantitative processing setting words or phrases. The first item must always be the no non-quantitative processing setting and subsequent items as appropriate to the sonar and data(e.g. "no_non_quantitative_processing simrad_noise_filter_weak simrad_noise_filter_medium simrad_noise_filter_strong").
short :flag_values		List of unique values (e.g. 0, 1, 3, 4) that indicate different non-quantitative processing settings that could be present in the sonar data. Must have the same number of values as settings given in the flag_meanings attribute.
:long_name = "Presence or not of non-quantitative processing applied to the backscattering data (sonar specific)"		
float receiver_sensitivity(ping_time, beam)	MA	Sensitivity of the sonar receiver for the current ping. Necessary for type 2 conversion equation.
:long_name = "Receiver sensitivity"		
:units = "dB re 1/ μ Pa"		
float sample_interval(ping_time)	M	Time between individual samples along a beam. Common for all beams in a ping.
:long_name = "Interval between recorded raw data samples"		
:units = "s"		
float :valid_min = 0.0		

float sample_time_offset(ping_time)	M	Time offset applied to sample time-stamps and intended for applying a range correction (e.g. as caused by signal processing delays). Positive values reduce the calculated range to a sample.
:long_name = "Time offset that is subtracted from the timestamp of each sample"		
:units = "s"		
sample_time_varied_gain(ping_time)	MA	Time-varied gain (TVG) used for each ping. Should contain TVG coefficient vectors. Necessary for type 2 conversion equations.
:long_name = "Time-varied-gain coefficients"		
:units = "dB"		
float transducer_gain(ping_time, beam)	MA	Gain of the transducer beam. This is the parameter that is set from a calibration exercise. Necessary for conversion equation type 1.
:long_name = "Gain of transducer"		
:units = "dB"		
float transmit_bandwidth(ping_time)	O	Estimated bandwidth of the transmitted pulse. For CW pulses, this is a function of the pulse duration and frequency. For FM pulses, this will be close to the difference between transmit_frequency_start and transmit_frequency_stop.
:long_name = "Nominal bandwidth of transmitted pulse"		
:units = "Hz"		
float :valid_min = 0.0		
float transmit_duration_equivalent(ping_time)	MA	Equivalent duration of the transmit pulse. This is the square pulse containing the same energy as the actual transmitted pulse. Necessary for both type 1 and 2 conversion equations.
:long_name = "Equivalent duration of transmitted pulse"		
:units = "s"		
float :valid_min = 0.0		

float transmit_duration_nominal (ping_time)	M	Nominal duration of the transmit pulse. This is not the equivalent pulse duration.
:long_name = "Nominal duration of transmitted pulse"		
:units = "s"		
float :valid_min = 0.0		
float transmit_frequency_start (ping_time, beam)	M	Frequency at the start of the transmit pulse. The beam dimension can be omitted, in which case the value applies to all beams in the ping.
:long_name = "Start frequency in transmitted pulse"		
:standard_name = "sound_frequency"		
:units = "Hz"		
float :valid_min = 0.0		
float transmit_frequency_stop (ping_time, beam)	M	Frequency at the end of the transmit pulse. The beam dimension can be omitted, in which case the value applies to all beams in the ping.
:long_name = "Stop frequency in transmitted pulse"		
:standard_name = "sound_frequency"		
:units = "Hz"		
float :valid_min = 0.0		
float transmit_power(ping_time)	MA	Electrical transmit power used for the ping. Necessary for type 1 conversion equations
:long_name = "Nominal transmit power"		
:units = "W"		
float :valid_min = 0.0		
float transmit_source_level (ping_time)	MA	Source level generated by the transmit ping. Necessary for type 2 conversion equations.
:long_name = "Transmit source level"		
:units = "dB re 1 μ Pa at 1m"		
transmit_t transmit_type(ping_time)	M	Type of transmit pulse.
:long_name = "Type of transmitted pulse"		

2.10.7 Vendor specific group

The vendor specific group contains information about the sonar and data specific to the sonar. Data in this group must not be necessary for normal quantitative use of the sonar data. The contents of this group are at the discretion of the sonar and software that writes the SONAR-netCDF4 file. The netCDF4 group name is **/Vendor_specific**. Currently, there is no mandatory information for the vendor specific group.

3 Conversion equations

This section provides detailed formulae on how to convert the backscatter data in the Sonar group into calibrated target strength and volume backscatter strength.

3.1 Type 1

Type 1 conversion equations are used for data recorded by the Simrad SU90, SX90, and SH90 omnisonars and are presented in detail by Macaulay *et al.* (2016).

The complex-valued backscatter data are converted into calibrated target strength via:

$$TS = 10 \log_{10}(P_r) + 40 \log_{10}(r) + 2\alpha r - 10 \log_{10}\left(\frac{P_t \lambda^2}{16\pi^2}\right) - G - 40 \log_{10}(\cos \gamma), \quad (1)$$

where P_r is linearly proportional to the received power (square of the magnitude of the complex number given by backscatter_r and backscatter_i, W) and r is the range between the transducer and target, calculated from:

$$r = \frac{c(dt \cdot i - t_0)}{2} \quad (2)$$

where c is sound speed (NetCDF4 variable is sound_speed_indicative, m/s), dt is the time between recorded samples (sample_interval, s), i is the sample number (from zero to one less than the number of samples), and t_0 is a time-offset (sample_time_offset, s).

The absorption coefficient of sound in water is α (absorption_indicative, dB/m), P_t is the transmit power (transmit_power, W), λ is the acoustic wavelength (derived from the average of transmit_frequency_start and transmit_frequency_stop, and sound_speed_indicative, m), G is the transducer gain (transducer_gain, dB), and γ is the beam tilt angle (derived from beam_direction_x, beam_direction_y, beam_direction_z, degrees from horizontal).

The volume backscatter strength (S_v , MacLennan *et al.*, 2002) is derived from a similar equation:

$$S_v = 10 \log_{10}(P_r) + 20 \log_{10}(r) + 2\alpha r - 10 \log_{10}\left(\frac{P_t \lambda^2 c \psi \tau_e}{32\pi^2}\right) - G - 40 \log_{10}(\cos \gamma) \quad (3)$$

where ψ is the equivalent beam angle (equivalent_beam_angle, sr), τ_e is the effective pulse duration (transmit_duration_equivalent, s), and G is the transducer gain (transducer_gain, dB).

3.2 Type 2

Type 2 conversion equations are intended for data recorded by Furuno omnisonars. The received real-valued backscatter data are converted into calibrated target strength via:

$$TS = 20 \log_{10}\left(\frac{A}{\sqrt{2}}\right) + 40 \log_{10}(r) + 2\alpha r - (SL + K + \Delta G + G_T) \quad (4)$$

where A (backscatter_r, 1) is linearly proportional to the amplitude of the received echo, r is the range as given by equation (2), and α is the absorption coefficient of

sound in water (absorption_indicative, dB/m). SL is source level (transmit_source_level, dB re $1\mu\text{Pa}$ at 1 m) and K is receiver sensitivity (receiver_sensitivity, dB re $1/\mu\text{Pa}$). Both parameters may depend on tilt angle of the beam for transmitting and receiving.

Gain correction, ΔG , (gain_correction, dB) is determined by the calibration. Time-varied gain, G_T (time_varied_gain, dB), is given at each sample number i .

The volume backscatter strength is derived from a similar equation:

$$S_v = 20\log_{10}\left(\frac{A}{\sqrt{2}}\right) + 20\log_{10}(r) + 2\alpha r - 10\log_{10}\left(\frac{c\tau_e}{2}\psi\right) - (SL + K + \Delta G + G_T) \quad (5)$$

where c is sound speed (sound_speed_indicative, m/s), τ_e is the effective pulse duration (transmit_duration_equivalent, s), and ψ is the equivalent beam angle (equivalent_beam_angle, sr). Considering the time-varied-gain (TVG) effect on the echo shape (Sawada and Furusawa, 1993), the replacement of r by $r - r_0$, where $r_0 = \frac{1}{4}c\tau_e$, is recommended (MacLennan, 1986; Furusawa *et al.*, 1999) for calculation of S_v .

4 Coordinate systems

4.1 Platform coordinate system

The coordinate system for the platform uses the right-handed Cartesian convention (Figure 1) with:

- x -axis parallel to the main axis of the platform, positive values toward the front of the platform (e.g. toward the bow of a ship),
- y -axis perpendicular to the main axis of the platform, positive values to the starboard side of the platform,
- z -axis vertical, positive values down from the platform,
- the origin is the common reference point where all three axes meet and is arbitrary.

Roll is positive with port side up, pitch is positive with bow up, and heading/yaw is positive clockwise. More specifically:

- looking along the positive x -axis, a positive rotation (roll) is clockwise (to starboard),
- looking along the positive y -axis, a positive rotation (pitch) is clockwise (bow up),
- looking along the positive z -axis, a positive rotation (yaw) is clockwise (to starboard).

The platform-heading variable (degrees clockwise from north) can be used to obtain the sonar orientation in world coordinates. This applies to both stationary and mobile sonar platforms.

The orientation of the platform is represented using the z - y' - x'' Tait-Bryan intrinsic rotation convention (en.wikipedia.org/wiki/Euler_angles), corresponding to yaw, pitch, and roll, respectively. Intrinsic means that rotations about the y -axis are measured after any rotation about the z -axis, and rotations about the x -axis are measured after rotations about the y -axis (for comparison, extrinsic angles are applied relative to a fixed-platform orientation). This is the most used rotation convention in the maritime field, and the main effect is that roll is measured relative to the plane tilted by the pitch angle.

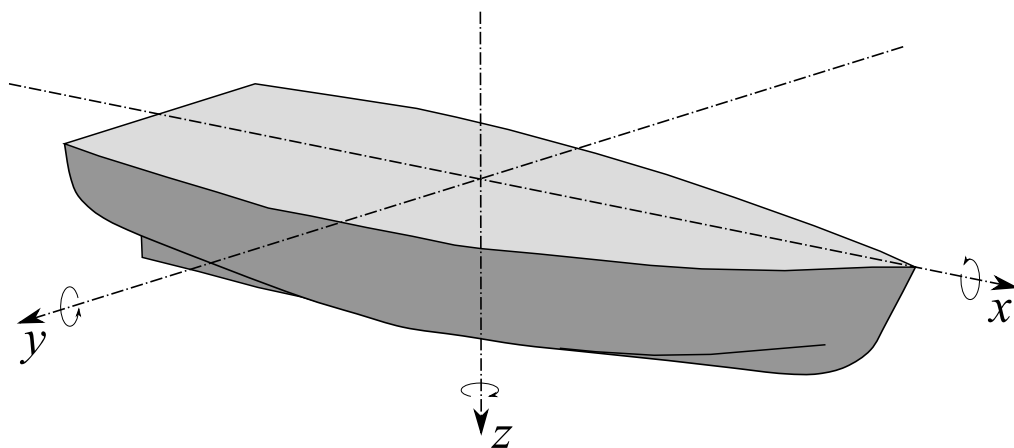


Figure 1. Platform coordinate system. The arrows on the axes indicate the positive direction and the arrows around each axis indicate positive rotations (Hull drawing based on image from Simrad SN90 manual, redrawn with permission).

4.2 Sonar beam coordinate system

The coordinate system for beam direction uses the same convention as the platform coordinate system, but with the origin at the transducer. Sonar beams can be compensated or uncompensated for platform motion. If uncompensated, the sonar beam coordinate system follows the yaw, pitch, and roll of the platform. If beams are compensated, the sonar beam coordinate system is translated and rotated per ping so that the y -axis is always horizontal and the z -axis is always vertical. The x -axis is always parallel to the platform's x -axis. The beam stabilisation variable indicates whether the beams are compensated or uncompensated.

4.3 Coordinate systems offsets and rotations

Several coordinate system offsets can be given in the Platform group. These allow for precise specification of the origin of sensors, such as the position and attitude sensors, and sonar transducer. The offset is a (x, y, z) tuple in the platform coordinate system. Offsets are to be interpreted as a vector that starts at the platform coordinate system origin and ends at the sensor position. For example, an offset of $(1, 2, -3)$ indicates a position that is 1 m toward the bow, 2 m to starboard, and 3 m above the origin of the platform coordinate system.

Some sensors (e.g. the attitude sensor) can have a rotation relative to the platform-coordinate system. This is represented as the extrinsic rotation required from the platform coordinate system's x -, y -, and z -axes to arrive at the sensor's x -, y -, and z -axes.

5 Revision history

Document version	SONAR-netCDF4 version	Date	Changes
1.7	1.0	29 May 2018	Modifications due to ICES CRR review and editorial process.
1.6	1.0	7 February 2018	Modifications due to further Topic Group input.
1.5	1.0	20 December 2017	Formatted to ICES CRR style.
1.4	1.0	18 September 2017	Added additional CF attributes to time variables. From Topic Group input: incorporation of Type 2 equation and additional variables to support it. Consistency and clarity corrections.
1.3	1.0	21 June 2017	Extensive and significant modifications derived from feedback on v1.2, from creation of test SONAR-HDF4 files, and from implementation of reading in LSSS.
1.2	1.0	2 February 2017	Further modifications after internal review.
1.1	1.0	13 January 2017	Modifications after generation of test datasets.
1.0	1.0	22 December 2016	Draft version for distribution to ICES WGFASSTopic Group on “Defining a data format for omnifisheries sonar”.

6 Acknowledgements

We are grateful for the thorough and considered contributions from the Marine Acoustics Society of Japan's Technical Committee. Advice and experience on the practicalities of reading netCDF files in analysis software was generously provided by Christian Michelsen Research AS, Echoview Software, and Nortek AS. Sindre Vatnehol and Arne Johannes Holmin kindly developed software to generate SONAR-netCDF4 files. We also thank Elodie Fernandez and Jim Biard for their invaluable reviews and suggestions for improving both the convention and this document.

7 References

- Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S., Blower, J., Caron, J., *et al.* 2017. NetCDF Climate and Forecast (CF) Metadata Conventions, version 1.7. 151 pp.
- Furusawa, M., Hamada, M., and Aoyama, C. 1999. Near range errors in sound scattering measurements of fish. *Fisheries Science*, 65: 109–116.
- Gee, L., Doucet, M., Parker, D., Weber, T., and Beaudoin, J. 2012. Is multibeam water column data really worth the disk space? *In* Taking care of the sea, pp. 81–86. Ed. by L. Dorst, T. van Dijk, R. van Ree, J. Boers, W. Brink, N. Kinneging, V. van Lancker, *et al.* Conference Proceedings of Hydro12, 12–15 November 2012, Rotterdam, The Netherlands. Hydrographic Society Benelux. 355 pp.
- ICES. 2016. A metadata convention for processed acoustic data from active acoustic systems. SISP 4 TG-AcMeta Version 1.10, ICES WGFASST Topic Group, TG-AcMeta. 47 pp.
- ISO. 2014. ISO 19115-1. Geographic information -- Metadata -- Part 1: Fundamentals. International Standard. 167 pp.
- Jackson, M. A., Groeber, M. A., Uchic, M. D., Rowenhorst, D. J., and Graef, M. D. 2014. h5ebds: an archival data format for electron back-scatter diffraction data sets. *Integrating Materials and Manufacturing Innovation*, 3: 4. <https://doi.org/10.1186/2193-9772-3-4>.
- Macaulay, G. J., Vatnehol, S., Gammelsæter, O. B., Peña, H., and Ona, E. 2016. Practical calibration of ship-mounted omni-directional fisheries sonars. *Methods in Oceanography*, 17: 206–220. <https://doi.org/10.1016/j.mio.2016.10.001>.
- MacLennan, D. N. 1986. Time varied gain functions for pulsed sonars. *Journal of Sound and Vibration*, 110: 511–522. [https://doi.org/10.1016/S0022-460X\(86\)80151-1](https://doi.org/10.1016/S0022-460X(86)80151-1).
- MacLennan, D. N., Fernandes, P., and Dalen, J. 2002. A consistent approach to definitions and symbols in fisheries acoustics. *ICES Journal of Marine Science*, 59: 365–369. <https://doi.org/10.1006/jmsc.2001.1158>.
- McQuinn, I. H., Reid, D. G., Berger, L., Diner, N., Heatley, D., Higginbottom, I., Andersen, L. N., *et al.* 2005. Description of the ICES HAC Standard Data Exchange Format, Version 1.60. ICES Cooperative Research Report No. 278. 84 pp.
- Sawada, K., and Furusawa, M. 1993. Precision calibration of echo sounder by integration of standard sphere echoes. *Journal of the Acoustical Society of Japan*, 14: 243–249. <https://doi.org/10.1250/ast.14.243>.
- The HDF Group. 2017. Hierarchical Data Format, version 5. <http://www.hdfgroup.org/HDF5>.
- Triton. 2013. Triton Imaging, Inc. eXtended Triton Format (XTF) Revision 35. 44 pp.
- UCAR. 2014. UDUNITS. <http://www.unidata.ucar.edu/software/udunits>.
- Unidata. 2017. Network Common Data Form (netCDF) version 4. UCAR/Unidata.

Annex 1: Working with netCDF4 files

NetCDF4 files are not commonly used for fisheries acoustics data. To facilitate the use of such files, this section provides simple examples of how to access and use SONAR-netCDF4 files in commonly used programming languages, namely Python, R, and Matlab.

Python example

```
#import packages needed to read and view data from netcdf4
from netCDF4 import Dataset
import numpy as np
import matplotlib.pyplot as plt

#Name of the netcdf file
filename = 'SU90-D20171107-T195023.nc'

#open the file
dataset = Dataset(filename)

#Open the group where the backscatter data is located
SonarGr = dataset.groups['Sonar'].groups['Beam_group2']

#Get the backscatter data from the 10th ping and
#and 31st beam
back_r = SonarGr.variables['backscatter_r'][9,30]
back_i = SonarGr.variables['backscatter_i'][9,30]

#close dataset
dataset.close()

#Compute the power
power = abs(np.vectorize(complex)(back_r,back_i))**2

#Plot the power of beam
plt.figure(1)
plt.clf()
plt.plot(10*np.log10(power))
plt.xlabel('Sample number')
plt.ylabel('Amplitude [dB]')
```

R example

```
library(h5)
SU90_nc <- PATH_TO_SONAR_NETCDF4_FILE

# The h5 package is able to read the example file:
data <- h5file(SU90_nc)
# Show the contents:
data

# List the dimensions of the data sets:
ds <- list.datasets(data)
dims <- lapply(ds, function(x) openDataSet(data, x)@dim)
names(dims) <- ds
dims

# Note that the backscatter data are stored as variable
# length datatype, so the dimensions in 'dims' does not
# reflect to the full size of the backscatter data.
# In this particular file the length of all beams and all
# pings of both Beam_group1 and Beam_group2 are identical:
bs_r1_name <- "/Sonar/Beam_group2/backscatter_r"
bs_i1_name <- "/Sonar/Beam_group2/backscatter_i"

# Save the real part and set dimension to
# [length of beams, number of beams, number of pings]:
backscatter1 <- complex(
  real=unlist(data[bs_r1_name][10,31]),
  imaginary=unlist(data[bs_i1_name][10,31]))
power <- abs(backscatter1)^2

# Plot the first and second beam of the third ping:
plot(10*log10(power), type="l",
     ylab="Power", xlab="Sample number")

# Close the file:
h5close(data)
```

Matlab example

```
% Example Matlab script to load a SONAR-netCDF4 file using the
% high-level HDF5 functions.
```

```
% File to read
```

```
file = 'SU90-D20171107-T195023.nc';
```

```
% Ping and beam to read
```

```
pingNo = 10;
```

```
beamNo = 31;
```

```
% Read the selected ping/beam from the file
```

```
amp_r = h5read(file, '/Sonar/Beam_group2/backscatter_r', ...
    [beamNo pingNo], [1 1]);
```

```
amp_i = h5read(file, '/Sonar/Beam_group2/backscatter_i', ...
    [beamNo pingNo], [1 1]);
```

```
% Calculate the power values for the ping.
```

```
power = abs(complex(cell2mat(amp_r), cell2mat(amp_i))).^2;
```

```
% Plot the power
```

```
plot(10*log10(power))
```

```
xlabel('Sample number')
```

```
ylabel('Power (dB)')
```

Annex 2: Contact information

Editors

Gavin John Macaulay

Institute of Marine Research,
Norway

gavin.macaulay@hi.no

Héctor Peña

Institute of Marine Research,
Norway

hector.pena@hi.no

Members of the Topic Group on Defining a data format for omnidirectional fisheries sonar (part of ICES Working Group on Fisheries Acoustics, Science, and Technology)

Akira Okunishi

Furuno Electric Co. Ltd., Japan

akira.okunishi@furuno.co.jp

Christophe Cobieres

iXblue SAS, France

christophe.corbieres@ixblue.com

Arne Johannes Holmin

Institute of Marine Research, Norway

ArneJohannes.Holmin@hi.no

Dezhang Chu

National Oceanic and Atmospheric
Administration, USA

dezhang.chu@noaa.gov

Atle Totland

Institute of Marine Research, Norway

atle.totland@hi.no

Doug McGowen

Edgetech,
USA

Doug.McGowen@edgetech.com

Benoit Berges

Wageningen Marine Research,
The Netherlands

benoit.berges@wur.nl

Gavin John Macaulay

Institute of Marine Research,
Norway

gavin.macaulay@hi.no

Briony Hutton

Echoview Software, Australia

briony.hutton@echoview.com

Glen Rice

National Oceanic and Atmospheric
Administration, USA

glen.rice@noaa.gov

Héctor Peña

Institute of Marine Research,
Norway

hector.pena@hi.no

Leon Smith

Faroe Marine Research Institute, Faroe
Islands

leonsmit@hav.fo

Inge Christian Eliassen

Christian Michelsen Research Institute,
Norway

inge.eliassen@cmr.no

Mariano Gutiérrez

Agroceánica Consultores, Perú

msgtorero@gmail.com

Jan Arge Jacobsen

Faroe Marine Research Institute,
Faroe Islands

janarge@hav.fo

Nils Olav Handegard

Institute of Marine Research, Norway

nilsolav@hi.no

Kouichi Sawada

Japan Fisheries Research Agency,
Japan

ksawada@fra.affrc.go.jp

Sindre Vatnehol

Institute of Marine Research, Norway

sindre.vatnehol@hi.no

Koshi Haraguchi

Sonic Corporation,
Japan

koshi-haraguchi@u-sonic.co.jp

Stéphane Gauthier

Fisheries and Oceans Canada, Canada

stephane.gauthier@dfo-mpo.gc.ca

Lars Nonboe Andersen

Simrad, Kongsberg Maritime, Norway

lars.nonboe.andersen@simrad.com