Open Geospatial Consortium

Submission Date 2025-06-10

Approval Date 2025-09-16

Publication Date: 2026-02-06

External identifier of this OGC® document: http://www.opengis.net/doc/CS/bag

Internal reference number of this OGC® document: 25-018

Version: 2.0.1

Category: OGC® Community Standard

Editor: Open Navigation Surface Working Group

Bathymetric Attributed Grid (BAG) 2.0.1 Community Standard

Warning

This document is an OGC Member endorsed international Community standard. This Community standard was developed outside of the OGC and the originating party may continue to update their work; however, this document is fixed in content. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC® Community Standard
Document subtype:
Document stage: approved
Document language: English

Format Specification Document

Description of the Bathymetric Attributed Grid Object, version 2.0 Release 2.0.1 (2022-12-01)

The release history is here.

Contents

Introduction

Motivation

The design of the 'Navigation Surface' [1] concept [2, 3] envisioned a new structure for hydrographic product creation from basic surveys. Relying on a database to hold all the original data, processed into the form of grids of the best available representation of the true nature of the seafloor, the Navigation Surface concept enables extracting whatever data are required for a particular product and, through automatic manipulation and/or cartography, enables the hydrographer to construct a product suitable for a particular purpose.

Depending on the task at hand, this might be a uniform grid at the best available resolution (e.g., for flow modeling), or a hydrographic vector-based chart for safety of navigation. To make this possible, the database must contain the best available data for each area, at the highest achievable resolution, and may contain data from a number of different sources, potentially from a number of different software packages.

To realize such a database, there is a need for a uniform file format that allows data to be passed between software packages, and between agencies involved in the collection, processing and dissemination of the data, while maintaining the integrity of the data and metadata at all times. The file format should also be sufficiently flexible to support processing of data without format conversion where possible, since format conversion can be an extremely expensive process. The Open

Navigation Surface (ONS) Project has as its mandate the task of building such a data file format, and developing and maintaining the source code for a software library to read and write this format so that adoption of the technology is eased for any developer. The source code library is developed on the Open Source model [4] so that the source code is freely available; all members of the ONS Working Group (ONSWG) and their respective employers have provided their effort on this basis.

This document describes the BAG format and the conduct of the ONSWG and its derivatives that maintain the BAG format reference implementation. For information on using the BAG reference implementation, please see the How-to Guide.

Nomenclature

The term "Navigation Surface" was coined to describe the combination of a data model representing the bathymetry and associated uncertainty, and the methods by which such objects could be manipulated, combined and used for a number of tasks, including products in support of safety of navigation. These multiple goals have led to some uncertainty about what exactly constitutes a Navigation Surface. To avoid any further confusion, a revised nomenclature has been designed.

In the ONS model, a unit of bathymetry is termed a Bathymetric Attributed Grid (BAG). A single BAG object represents one contiguous area of the skin of the Earth at a single resolution but can represent data at any stage of the process from raw grid to final product. The name Navigation Surface (NS) is reserved for a final product BAG destined specifically for safety-of-navigation purposes. The status of any particular BAG is distinguished solely by the certification section of metadata embedded in the file.

Properties of the BAG

The Navigation Surface concept requires that in addition to estimation of depth, an estimate of the uncertainty associated with the depth must be computed and preserved. In order to make the system suitable to support safety of navigation applications, there is a means to over-ride any automatically constructed depth estimates with "Hydrographer Privilege", (essentially, a means to specify directly the depth determined by a human observer as being the most significant in the area, irrespective of any statistical evidence to the contrary).

There is also the requirement to provide data on the data (i.e., metadata, which describe all aspects of the data’s life, from methods of capture to processing methods, and from geospatial extents to responsible party).

Finally, there must be a means to certify that the data in the file has been inspected by someone with appropriate experience and authority, that the data have been verified as suitable for some specific purpose, and that the file has not changed since this certification was made: in essence, a digital replacement for the Hydrographer’s signature.

Means to incorporate all of these requirements in a portable, extensible, platform- and vendor-neutral format are provided in the remainder of this Format Specification Document (FSD).

Status of the Project

The ONSWG maintain both a website, which contains the current recommended release of the source code and supporting documentation, and a Git repository, which contains the source code of the BAG library.

The BAG library is designed to be built from source code, including all component libraries that the BAG uses. All component libraries are covered under open-source licenses, although not all the licenses are alike. Users should ensure that their use of the BAG library, and its dependencies, is compliant with the terms of the appropriate license before using the component libraries. The ONS source code library itself is licensed under the 3-Clause BSD License and you are free to modify, adapt and otherwise reuse the source code, including the construction of derivative products based on the source code, so long as you do so in accordance with the terms of the 3-Clause BSD license. However, if a bug is found or there is a way to improve part of the library, it is requested that this be communicated to the ONSWG as described in the BAG Format Extensions chapter. Extensions to the library shall not be added without permission of the BAG Architecture Review Board. Applications for extensions are also covered in the BAG Format Extensions chapter.

History

The idea for what became the Open Navigation Surface Project had been discussed within the US hydrographic community for some time, but first crystallized in late 2003 through comments of M. Paton at IVS3D Ltd., which were adopted by B. Calder at CCOM/JHC with the intent of acting as an independent third-party broker for the development of the library. The ONSWG first met in early 2004, and outlined the requirements for the BAG file structure, the ethos of the project and the basic functionality of the source code library. A presentation on the structure of the project was made at US Hydro 2005 in March, and development continued until the second ONSWG meeting in mid-2005, when the majority of the first release of the library was pulled together in a little under a week. A second presentation, including demonstrations of the library linked into SAIC SABER, CARIS HIPS and IVS Fledermaus, was given at Shallow Survey 2005 in September and development of the library continued using e-mail and telephone conferencing until the first Candidate Release on 8 February 2006. Following comments from users and further development, the first official release of the library was 8 April 2006.

Identification

This document describes the Bathymetric Attributed Grid file format, version 2.0.1. This version corresponds to the Git release tag “release-2.0.1”, which may be obtained from the Open Navigation Surface (ONS) Project on GitHub.

Next: Overview of the BAG Structure

Overview of the BAG Structure

Top-Level Requirements

Several top-level requirements for what would become the BAG file format were identified during the inaugural ONSWG meeting early in 2004. These requirements provided the foundation on which many of the architectural decisions have been made. In particular, the ONSWG adopted the following high level requirements:

1. A freely distributable source code library shall be available for the BAG.
2. BAG shall adhere to a defined format, expressed via the source code library and accompanying documentation.
3. The ONSWG shall provide and oversee a change process to allow for incorporation of bug fixes to the library and extensions to the format.
4. The BAG library and file format shall be operating system independent, and supported on at least the Win32, and Linux operating systems.
5. The BAG library and file format shall support files larger than 2 gigabytes.
6. The BAG file format shall include mandatory data elements, and optional extensions.
7. The BAG mandatory elements include metadata, elevation grid, vertical uncertainty grid, change tracking list, and digital signature.
8. The BAG file format and access library shall maintain all mandatory data elements within a single, self-contained, file structure.
9. The BAG digital signature shall provide a mechanism to describe intended use of the dataset.
10. The BAG digital signature shall provide a mechanism to validate that the contents of the file have not changed since the original signature.
11. Extensions to the BAG file format shall be fully defined and approved by the ONSWG prior to adoption.
12. The ONSWG shall maintain a website that describes the status of the project and allows for releases of the library to be downloaded.
13. The ONSWG shall maintain a revision control system to manage configuration baselines of the BAG library software and documentation.

BAG File Layout

Fig. 1.1 provides a conceptual view of the contents of each of the mandatory data elements of a BAG. The elevation grid contains a two-dimensional array of bathymetry data. The uncertainty grid is co-located with the elevation grid and contains a two-dimensional array describing the vertical uncertainty of the elevation grid. The tracking list details hydrographer modifications, and the certification section specifies authenticity and intended use of data.

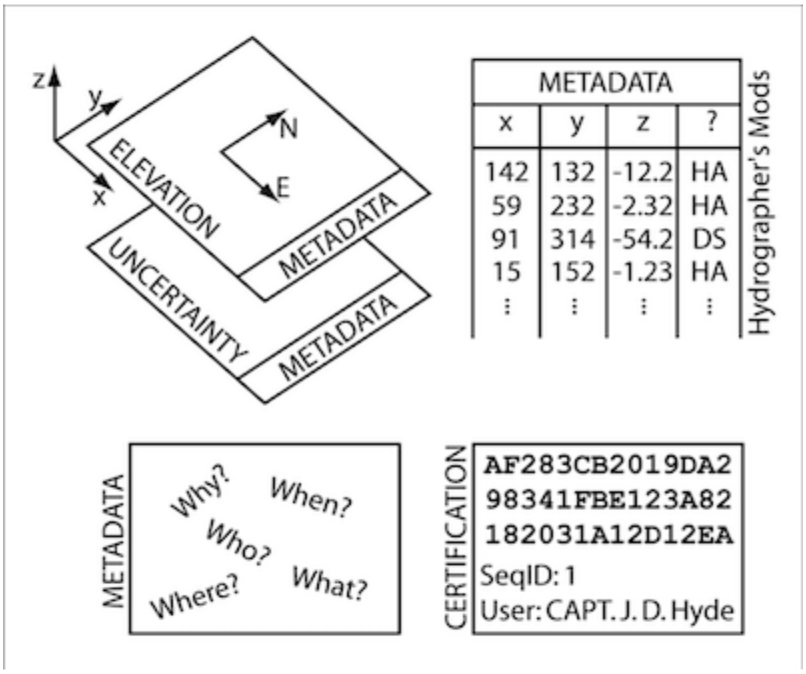

Figure 1.1: Mandatory elements of a BAG.

Fig. 1.2 shows the structural layout of the mandatory elements of the BAG. The version tag is a simple text string that specifies the version of the format at the time the BAG was created. The metadata, elevation, uncertainty, and change-list are each HDF5 datasets with a dataset substructure appropriate for the information being stored. The signature section is simply a byte stream appended after the end of the HDF5 file using standard C file access mechanisms. The rationale for adding the signature byte stream after the end of the HDF5 file is to ensure that the contents of the signature do not modify the file that it is trying to protect.

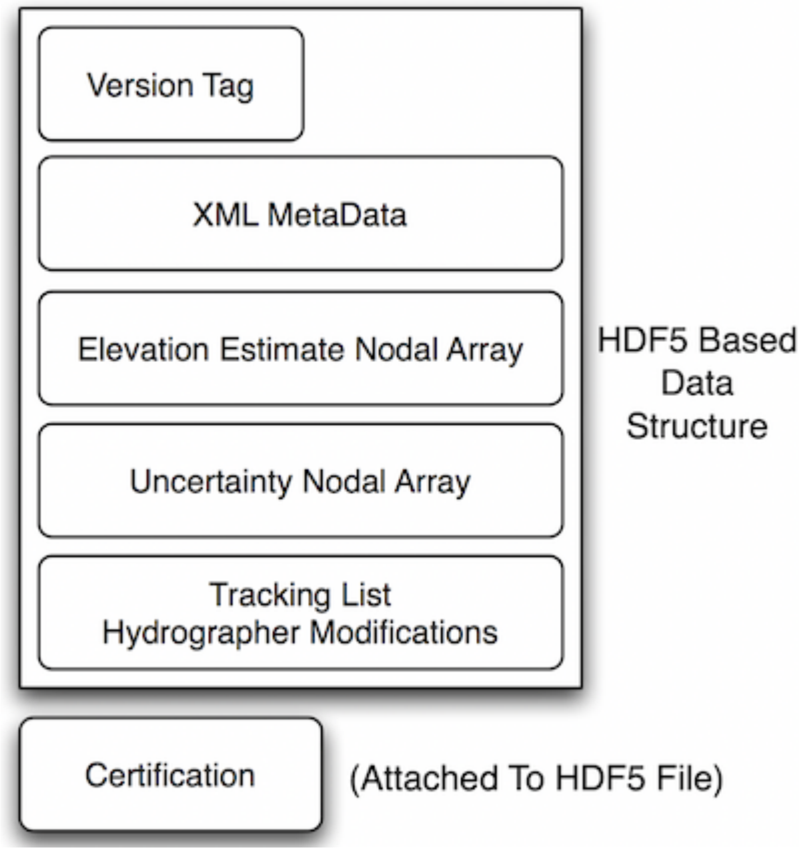

Figure 1.2: BAG file structure.

For BAG version 1.1.0, optional datasets for the addition of optional surface layers were added and are shown in Fig. 1.5 (see Nominal Elevation section below), for the addition of optional surface layers. These optional surface layers are each HDF5 datasets defined at the root level of the BAG and follow the same substructure as the elevation and uncertainty datasets. Further extensions for auxiliary information on depth reconstructions and variable resolution layers (from release 1.6.0) have also been defined.

From version 1.1.0, a BAG has a single fixed node-spacing (referred to as grid resolution) and represents a contiguous region of the surface of the earth. A future version of BAG may support storage of multiple grids. For all practical purposes, grid size is constrained only by available disk space. Row and column values are dimensioned as unsigned 32-bit integers, and HDF5 places no fixed upper limit on row or column dimension or on file size. The Axiomatic Definitions chapter details the coordinate system and standard units of measure used by the BAG library.

Metadata

Requirements for Metadata

With the evolution of digital data, also comes the necessity to accurately track and attribute this data. BAG is no exception. One of the primary goals behind the BAG initiative was to provide a way to uniformly encode, decode and exchange the information about who, what, when, where, and how the BAG file was created.

Standard Object Model

During the first meeting of the Open Navigation Surface Working Group in January 2004, it was concluded that the ISO metadata standards 19115 and 19115-2 would be used as the common framework to store the metadata information needed by the BAG format. A sub-working group was selected and assigned to work on creating the appropriate metadata profile to describe the BAG file contents. The profile was created and is depicted in Fig. 1.3. This profile describes the minimum required fields for a valid BAG file. However, the user is not limited to this profile, and may include any of the other components described in the 19115 Standard.

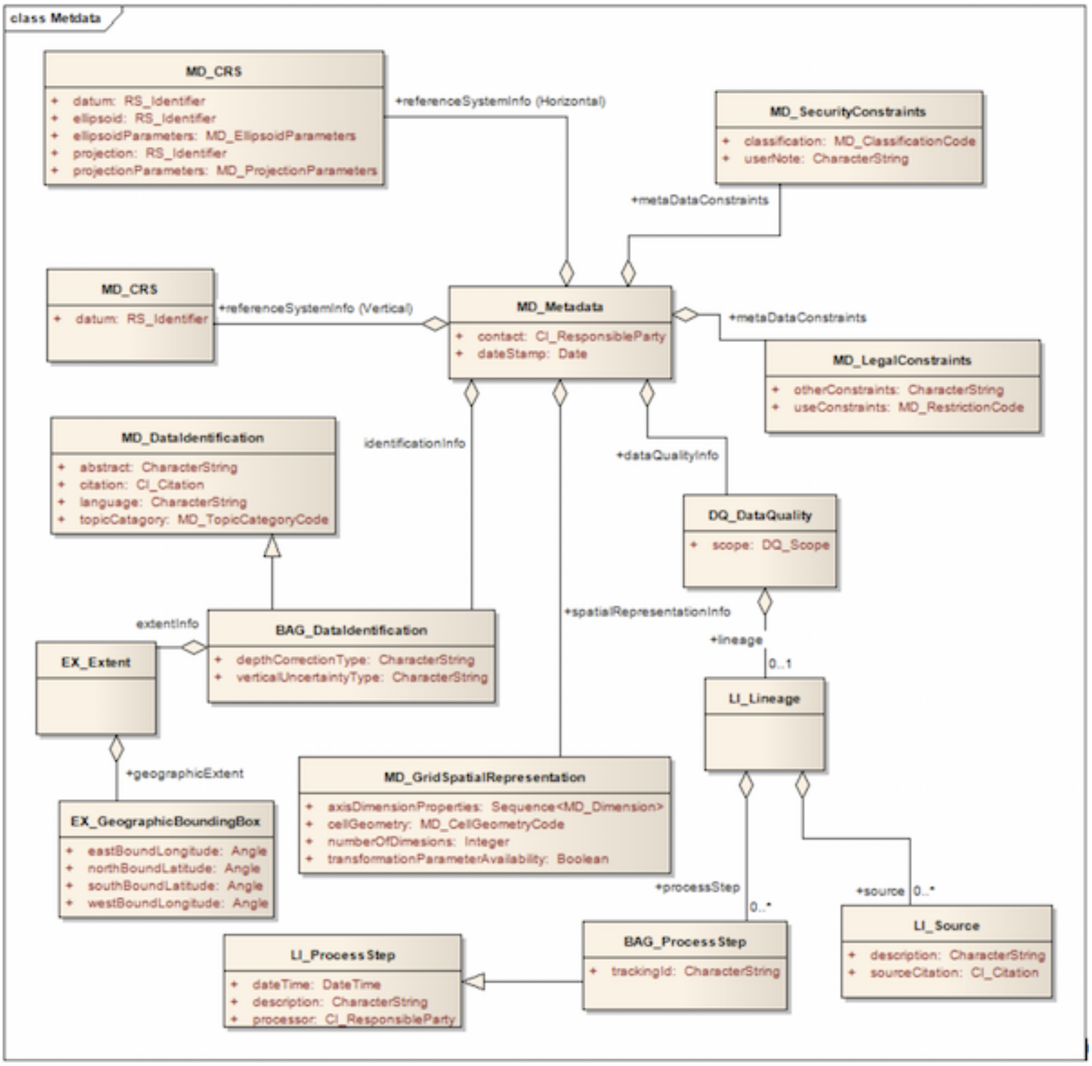

Figure 1.3: UML model for BAG metadata profile.

The ISO 19115 object model provided most of the needed classes and attributes required to fully describe the BAG content. There were three extensions made to the model to support detailed description of the data. First, the `verticalUncertaintyType` attribute was added to allow the BAG file to accurately describe the source and meaning of the encoded `Uncertainty` grid layer. Table 1.1 shows the list of supported values as of the 1.1 release.

Table 1.1: Table of uncertainty types.

Enumerated Value	Description
rawStdDev	"Raw Standard Deviation" - Raw standard deviation of soundings that contributed to the node
cubeStdDev	"CUBE Standard Deviation" - Standard deviation of soundings captured by a CUBE hypothesis (i.e., CUBE's standard output of uncertainty)
productUncert	"Product Uncertainty" - NOAA standard product uncertainty V1.0 (a blend of CUBE uncertainty and other measures).
noaaProduct_2024	"Product Uncertainty 2024" - NOAA standard product uncertainty version 2024 (https://repository.library.noaa.gov/view/noaa/61830)
historicalStdDev	"Historical Standard Deviation" – Estimated standard deviation based on historical/archive data.
averageTPE	"Average TPE" - Average of all of the contributing sounding TPE's within the node.
unknown	"Unknown" - The uncertainty layer is an unknown type

Secondly, the `trackingId` extension allows internal Tracking List entries to be associated with a unique entry in the metadata so that the changes can be properly attributed, described and easily referenced.

Thirdly, the `depthCorrectionType` extension was added to allow the BAG file to accurately describe the correction performed on the data. Table 1.2 shows the list of supported values as of the 2.0 release.

Table 1.2: Table of depth correction types.

Enumerated Value	Description
trueDepth	"True Depth" – Depth corrected for sound velocity
nominalDepthMetre	"Nominal at 1500m/s"- Depth at assumed sound velocity of 1500m/s
nominalDepthFeet	"Nominal at 4800ft/s" – Depth at assumed sound velocity of 4800ft/s
correctedCarters	"Corrected via Carter's Tables" – Corrected depth using Carter's tables
correctedMatthews	"Corrected via Matthew's Tables" – Corrected depth using Matthew's tables
unknown	"Unknown" – Unknown depth correction type or mixture of above types

Common Encoding

After choosing the ISO standard object model for metadata encapsulation, a format was needed to store this structure. The ISO had already begun work on the 19139 Standard Geographic MetaData XML (gmd) encoding, an XML Schema implementation derived from ISO 19115. It was decided that XML seemed to be a logical medium as it was both machine- and human-readable, and is already a well-known storage format for many other standards.

For BAG implementations to be flexible, the metadata is provided to the library as a character stream. The stream is validated and stored in its own HDF5 block at the end of the file before the digital signature. This data block in the HDF5 file is configured as an extendable block to allow the metadata to grow in the future and not disturb the remaining data blocks. In a similar fashion, the metadata can be extracted from the file in its native XML character stream and thus can be processed by any regular XML tools.

Internal Parsing and Validation

Each time a metadata stream is processed by the library (for read, creation, or update) it must be validated to ensure that it is well-formed XML and, more importantly, meets the requirements of the defined BAG profile. In implementations of BAG, this validation should occur transparently to the user when a BAG file is opened or created.

Elevation Grid

The elevation section of the BAG contains a two-dimensional matrix of elevation values, organized in row major order, and starting from the south-western most data point, where each value is defined to be the elevation at an exactly specified geographic point (node).

An important distinction to note is the contrast between this node-based approach where each grid value describes the elevation exactly at a specified geographic point, as compared with a cell-oriented approach, where each grid value might describe the elevation as applying over the extents of the cell size. The elevation data in a BAG describe the elevation only at the exact node locations and offer no information about the elevation between the nodes.

The units of the elevation values are meters, and the sign convention is for z to be positive for values above the vertical datum. The reference vertical datum for the BAG is one of the mandatory Metadata items. This sign convention follows directly from the right-hand coordinate system definition that BAG adheres to. See Axiomatic Definitions for additional information. The elevation values are encoded as four byte (32-bit) floats within an HDF5 dataset. The x and y location of each elevation value is not explicitly saved within the dataset. Rather, the x, y location of the south-western most node is saved within the metadata section, along with the x node spacing and the y node spacing. Fixed, regular spacing between nodes is required. he unknown state for elevation is defined to be 1,000,000.0 (1.0e6).

Uncertainty Grid

The uncertainty section of the BAG contains a two-dimensional matrix of values that specify the vertical uncertainty at each node. The elevation grid and the uncertainty grid are explicitly co-aligned. The values are expressed as positive quantities in units of meters.

As detailed in Table 1.1, the uncertainty grid supports multiple definitions of vertical uncertainty. This allows BAGs to span the expected range of data products from raw, full resolution grid to final compiled product. For example, a BAG at the stage of final survey data processing should contain uncertainty information germane to the survey data itself and intended to be used for information compilation. A BAG intended for navigational purposes would need to specify the overall uncertainty to the mariner—-these two values for uncertainty may be quite different.

A recipient of a BAG file can refer to the uncertainty definition in the Metadata to gain an understanding of how the uncertainty was computed. The uncertainty values are encoded as four byte (32-bit) floats within an HDF5 dataset. The unknown state for uncertainty is defined to be 0.0.

Tracking List

The tracking list contains a simple list of the original elevation and uncertainty values from any node of the surface that has been modified to account for hydrographer overrides of the basic surface definition (e.g., as originally computed by an algorithmic method). The tracking list dataset and corresponding information contained in the metadata exist to provide an audit trail record of changes made to the data by manual intervention. The contents of the tracking list dataset are shown in Fig. 1.4.

Field	Datatype
row	U32
col	U32
depth	F32
uncertainty	F32
track_code	U8
list_series	U16

Figure 1.4: Tracking list item - field names and datatypes

The `row` and `col` fields describe the node that has been modified. The `depth` and `uncertainty` fields contain the original values prior to manual intervention. The `track_code` field provides a reason code for the manual modification of a grid surface node.

The `list_series` field of the tracking list item structure is used to identify an entire series of tracking list items and associate them with supplemental details in the BAG metadata. There should be corresponding metadata entries for each suite of list_series' tracking list items. One can then gather from the metadata the circumstances and motivation behind a particular group of edits to the BAG depth and uncertainty surfaces.

Extensions

Extensions are optional and not required for an HDF file to be qualified as a BAG. Vendors or other third parties can apply for extensions to the format by the method outlined in this requirements document (please refer to BAG Format Extensions).

Nominal Elevation

Note Fig. 1.5 shows an extension made to the BAG API, starting with version 1.5.1, regarding the addition of optional datasets. `Nominal_Elevation` is one of the optional layers that can be included in the BAG.

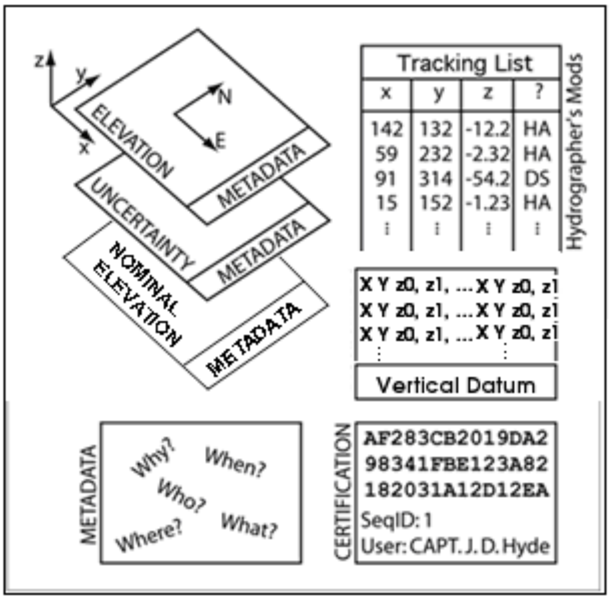

Figure 1.5: Example of BAG with optional dataset of nominal elevation.

`Nominal Elevation` is the term used by some hydrographic organizations to refer to an elevation value that is corrected for sound speed such that an echosounder (typically a single-beam system) that uses a fixed, assumed, speed of sound will indicate the same depth as that provided in the Nominal Elevation layer. Often, this is used where the echosounder cannot have a speed of sound profile applied, or where obtaining such a profile would be impossible, or prohibitively expensive. The speed of sound used for "nominal" should be documented in the metadata.

Vertical Surface Correctors

This optional dataset may comprise a regularly spaced grid topography or a set of irregularly spaced coordinates. The intended approach is that the BAG creator provides sufficient data density to support a simplistic inverse distance interpolation scheme to generate correction values at the full resolution of the BAG elevation nodes. The library includes an API function to perform this computation based on the surface correctors dataset and the desired BAG surface for either type of topography (regular or irregular spacing).

Elevation Solution Group

The elevation solution group is designed to contain the shoal elevation, standard deviation, and number of soundings associated with a node in the grid. The shoal elevation is the elevation value of the least-depth measurement selected from the subset of measurements that contributed to the elevation solution. The number of soundings is the number of elevation measurements selected from the subset of measurements that contributed to the elevation solution. The stddev is the standard deviation computed from all elevation values that contributed to the node. Note that the stddev value is computed from all measurements contributing to the node, whereas shoal elevation and number of soundings relate to the chosen elevation solution.

Node Group

The node group has two components: hypothesis strength and number of hypotheses. The hypothesis strength and number of hypotheses are computed in the CUBE/CHRT algorithm.

Georeferenced Metadata Raster Attribute Table

An optional georeferenced metadata raster attribute table is a raster and table pair where the raster value indicates the row of the table where a group of metadata is valid (Fig. 1.6). This construct requires definition of the metadata fields available in the table. Standard georeferenced metadata profile definitions can be found in Appendix A. There can be multiple metadata raster attribute tables, each associated with a given BAG layer (e.g., Elevation, Uncertainty, etc.) and each with its own distinct metadata profile. All metadata layers are found in a node under `BAG_root` called `georef_metadata`, and then each raster table pair is found in its own node with the name of the profile. The raster in the profile node is named `keys`, and shall match the shape of the required elevation layer. The no data value for this layer is 0, and the first row (zeroth row) of the table is ignored. The table in the profile node is named `values` and must have the rows indicated by the raster. Variable resolution keys are stored in an array the same length as, and corresponding to, `varres_refinements` with an index indicating the table row in the same fashion as the raster.

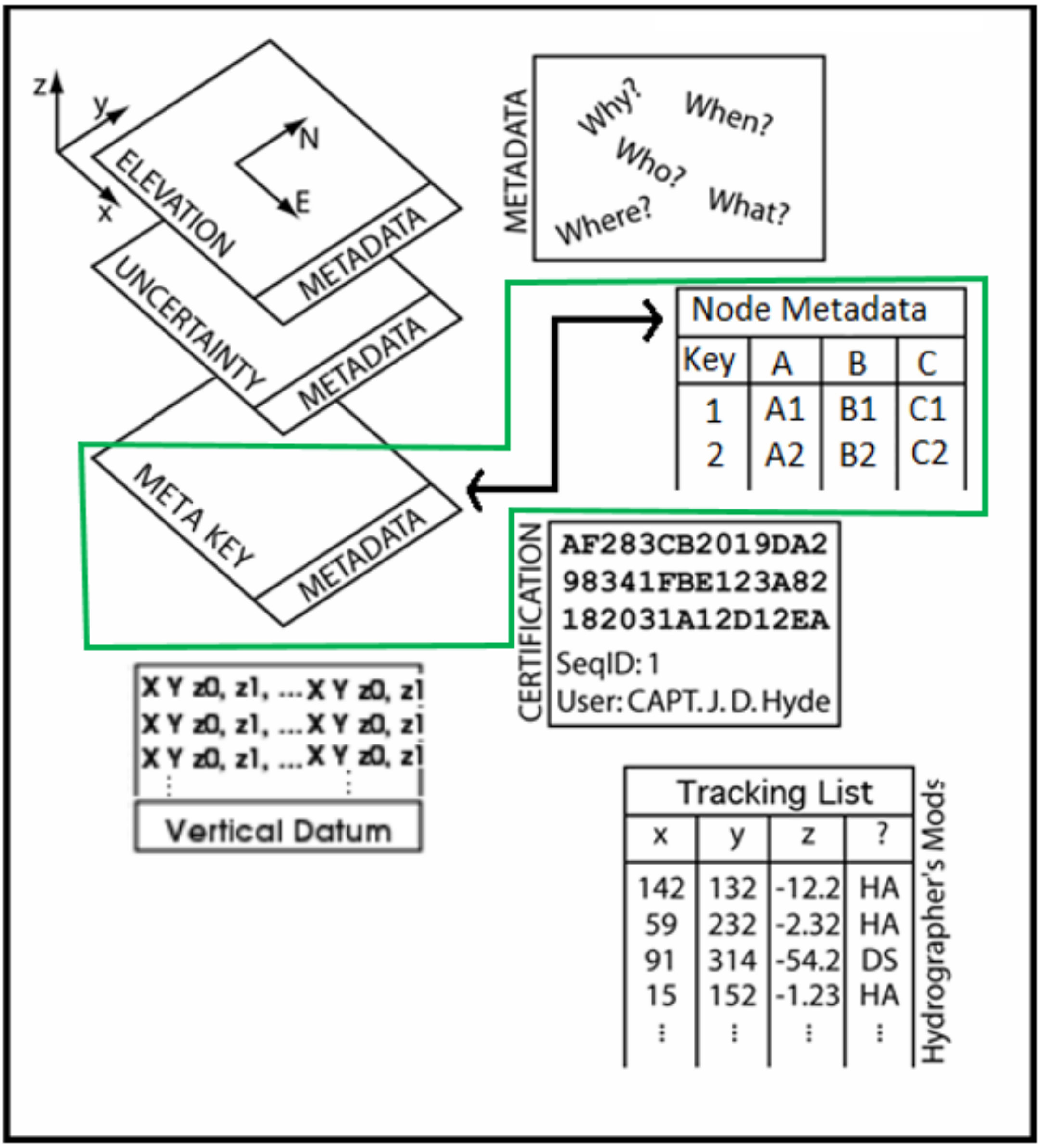

Figure 1.6: Example of BAG with optional georeferenced metadata.

Certification

> *Note: Due to the large-scale changes to the BAG codebase as part of moving from C to C++14, and deprecation of the original cryptographic library, support for BAG Digital Signatures has been removed from the 2.0.1 codebase. Signing support will be added back to a subsequent release of the BAG library. In the meantime, however, the rationale for, and content requirements of, the Digital Signature Scheme remain the same.*

Intent of the BAG Digital Signature Scheme

In a traditional hydrographic processing workflow, there is a strict chain of custody for all data that is to be used for nautical charting. At each stage of the chain, a responsible authority reviews the data and the processes applied to it, and certifies that the data is fit for some intended purpose. This may be that the data is ready for final plotting, that it is ready to be combined with other data in a compilation, or that the compilation is suitable as an aid to safe surface navigation. Generally, this is done by some physical signature on appropriate archival documentation, which is traditionally the hydrographic smooth sheet or fair sheet.

With an all-digital product, however, there is no opportunity to affix a physical signature to the data object. In addition, with a dense data object such as a BAG, the opportunity for single-bit errors in transmission to cause navigationally significant changes to the data which are otherwise undetectable is greatly increased. The Digital Signature Scheme (DSS) for the BAG is designed to provide an equivalent analogue for the physical hydrographer's signature, and to ensure that any modifications to the data, either by mistake or malicious action, are readily detectable.

This section describes the implementation for the DSS interaction with the BAG file format in outline. Full details of the process are available in the whitepaper, "Digital Signature Scheme for the Bathymetric Attributed Grid (BAG)" [5], which is available from the project's website.

DSS Implementation

The basic entity of the DSS is the Digital Signature (DS), a multi-byte sequence of digits computed from the contents of the BAG file excluding the certification information and another number, known as the secret key (SK), belonging to the person or entity signing the BAG, known as the Signature Authority (SA). The SK is known only to the SA, and as the name suggests should be kept confidential since knowledge of the SK would allow anyone to certify BAGs as if they were the SA. The DS value can be shown to be probabilistically unique for the contents of the BAG and the SK in the sense that, with vanishingly small probability, no two BAGs would generate the same DS with a particular SK, and no two SKs would generate the same DS with the same BAG.

Corresponding to the SK, there is a public key (PK) that can be distributed freely. There is no way to compute the DS using the PK. However, given a BAG and a DS purported to have been constructed with the SK, it is simple to verify whether the BAG has changed, or if another SK was used to construct the certification.

In addition to the basic DS required for the DSS, the BAG certification block contains a 32-bit integer used to link the certification event with an entry in the metadata's lineage section which describes the reasons for certification. The intent of this is to ensure that the user can provide suitably flexible descriptions of any conditions attached to the certification event, or the intended use of the data so certified. This 'Signature ID' shall be a file-unique sequentially constructed integer so that a certification block can be unambiguously associated with exactly one lineage element.

The DSS is not mandatory, in the sense that the API does not enforce checks for certification blocks or DS results as BAG files are opened, written or closed. A BAG without a certification block shall be considered valid. The BAG API provides means to construct and verify DS values, but does not address questions of key distribution, certificate generation or certificate signing. Users are urged to consult an appropriate reference (e.g., [6]) for details of these processes.

Structure of the BAG Certification Block

The BAG DS information shall be maintained in a certification block of length 1024 bytes, appended to the end of the HDF5 data. The structure of the certification block shall be as shown in Fig. 1.7. The ID number shall be a constant integer value (0x4F4E5343 or “ONSC”) used to identify the block, and the version byte shall be used to identify the structure of the remainder of the block between different versions of the algorithm. The SigID number corresponds to the Signature ID described above and shall be followed immediately by the DS values which shall be stored sequentially as a length byte followed by the digits of the element. The CRC-32 checksum shall be used to ensure that any accidental or intentional corruption of the certification block will be detectable. The block shall be stored in little endian format, and zero padded to the full size of the block.

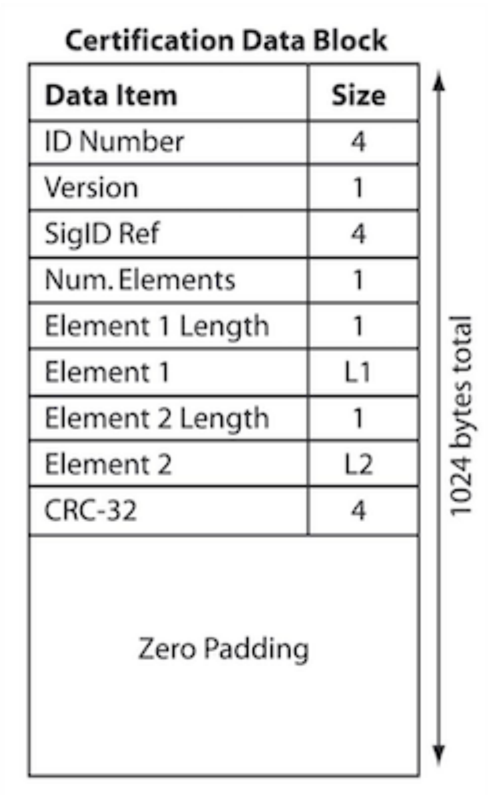

Figure 1.7: Structure of the BAG Digital Signature Scheme certification block.

<u>Next: Encapsulation</u>

Encapsulation

BAG data are encoded as <u>HDF5</u> files. HDF5 is a hierarchical data format product consisting of a data format specification and a supporting library implementation, it is also an OGC Community Standard. HDF5 provides two primary structures: groups and datasets. They are defined as:

- HDF5 Group: a grouping structure containing instances of zero or more groups or data sets, together with supporting metadata.

- HDF5 Dataset: a multidimensional array of data elements, together with supporting metadata or attributes.

An HDF5 “Group” provides the top-level structure for the data contents of a BAG. The major subcomponents are defined using the HDF5 “Dataset” types, and “Attribute” types. Within each “Dataset”, further structural decomposition is specified via the DATATYPE and DATASPACE parameters. “Attributes” are included were appropriate to provide “Dataset” specific metadata. The specific HDF5 type definitions that define the BAG encapsulation structure are illustrated in Fig. 2.1. Note that the digital signature is not shown in Fig. 2.1. As described in Structure of the BAG Certification Block, the digital signature byte stream is appended to the end of the HDF5 group.

Figure 2.1 shows the structure of BAG data encapsulated using HDF5:

```
$ h5dump -A examples/sample-data/sample.bag
HDF5 "examples/sample-data/sample.bag" {
GROUP "/" {
   GROUP "BAG_root" {
      ATTRIBUTE "Bag Version" {
         DATATYPE  H5T_STRING {
            STRSIZE 32;
            STRPAD H5T_STR_NULLTERM;
            CSET H5T_CSET_ASCII;
            CTYPE H5T_C_S1;
         }
         DATASPACE  SCALAR
         DATA {
         (0): "2.0.1"
         }
      }
      DATASET "elevation" {
         DATATYPE  H5T_IEEE_F32LE
         DATASPACE  SIMPLE { ( 100, 100 ) / ( 100, 100 ) }
         ATTRIBUTE "Maximum Elevation Value" {
            DATATYPE  H5T_IEEE_F32LE
            DATASPACE  SCALAR
            DATA {
            (0): 99.99
            }
         }
         ATTRIBUTE "Minimum Elevation Value" {
            DATATYPE  H5T_IEEE_F32LE
            DATASPACE  SCALAR
            DATA {
            (0): -10
            }
         }
      }
      DATASET "metadata" {
         DATATYPE  H5T_STRING {
            STRSIZE 1;
            STRPAD H5T_STR_NULLTERM;
            CSET H5T_CSET_ASCII;
```

```
      CTYPE H5T_C_S1;
   }
   DATASPACE  SIMPLE { ( 11005 ) / ( H5S_UNLIMITED ) }
}
DATASET "nominal_elevation" {
   DATATYPE  H5T_IEEE_F32LE
   DATASPACE  SIMPLE { ( 100, 100 ) / ( 100, 100 ) }
   ATTRIBUTE "max_value" {
      DATATYPE  H5T_IEEE_F32LE
      DATASPACE  SCALAR
      DATA {
      (0): 515.05
      }
   }
   ATTRIBUTE "min_value" {
      DATATYPE  H5T_IEEE_F32LE
      DATASPACE  SCALAR
      DATA {
      (0): 1
      }
   }
}
DATASET "tracking_list" {
   DATATYPE  H5T_COMPOUND {
      H5T_STD_U32LE "row";
      H5T_STD_U32LE "col";
      H5T_IEEE_F32LE "depth";
      H5T_IEEE_F32LE "uncertainty";
      H5T_STD_U8LE "track_code";
      H5T_STD_I16LE "list_series";
   }
   DATASPACE  SIMPLE { ( 0 ) / ( H5S_UNLIMITED ) }
   ATTRIBUTE "Tracking List Length" {
      DATATYPE  H5T_STD_U32LE
      DATASPACE  SCALAR
      DATA {
      (0): 0
      }
   }
}
DATASET "uncertainty" {
   DATATYPE  H5T_IEEE_F32LE
   DATASPACE  SIMPLE { ( 100, 100 ) / ( 100, 100 ) }
   ATTRIBUTE "Maximum Uncertainty Value" {
      DATATYPE  H5T_IEEE_F32LE
      DATASPACE  SCALAR
      DATA {
      (0): 100.01
      }
   }
   ATTRIBUTE "Minimum Uncertainty Value" {
      DATATYPE  H5T_IEEE_F32LE
      DATASPACE  SCALAR
      DATA {
```

```
            (0): 0
            }
         }
      }
      DATASET "vertical_datum_corrections" {
         DATATYPE  H5T_COMPOUND {
            H5T_IEEE_F64LE "x";
            H5T_IEEE_F64LE "y";
            H5T_ARRAY { [1][2] H5T_IEEE_F32LE } "z";
         }
         DATASPACE  SIMPLE { ( 3, 3 ) / ( H5S_UNLIMITED,
H5S_UNLIMITED ) }
         ATTRIBUTE "surface_type" {
            DATATYPE  H5T_STD_U8LE
            DATASPACE  SCALAR
            DATA {
            (0): 2
            }
         }
         ATTRIBUTE "vertical_datum" {
            DATATYPE  H5T_STRING {
               STRSIZE 1;
               STRPAD H5T_STR_NULLTERM;
               CSET H5T_CSET_ASCII;
               CTYPE H5T_C_S1;
            }
            DATASPACE  SIMPLE { ( 0 ) / ( 256 ) }
            DATA {
            }
         }
      }
   }
}
}
```

Figure 2.1: Structure of BAG data encapsulated using HDF5.

Table 2.1 defines the contents of the HDF data elements belonging to the `BAG_root` Group.

Table 2.1: Contents of BAG_Root group

Entity Name	Data Type	Domain	Required
BAG Version	String	Maximum 32 bytes available	Yes
metadata	Dataset	Detailed in table 4	Yes
elevation	Dataset	Detailed in table 5	Yes
uncertainty	Dataset	Detailed in table 6	Yes
tracking list	Dataset	Detailed in table 7, and in table 8	Yes
nominal_elevation	Dataset	Detailed in table 9	No
node	Dataset	Detailed in table 10	No
elevation_solution	Dataset	Detailed in table 11	No

Entity Name	Data Type	Domain	Required
georef_metadata	Dataset	Detailed in table 12	No

Table 2.2 defines the metadata items used within the BAG library. These items must be present and properly defined for BAG data to be read. Note that this listing of metadata items does not specify the mandatory metadata items required by the ISO 19115 Standard.

Table 2.2: Group level metadata - grid parameters.

Entity Name	XML Tag Nesting	Data Type	Domain
CoordSys			
Coordinate System code	Reference System Info/ projection/ Identifier/code	Non-Null String	Geodetic
			GEOREF
			Geocentric
			Local_Cartesian
			MGRS
			UTM
			UPS
			Albers_Equal_Area_Conic
			Azimuthal_Equidistant
			BNG
			Bonne
			Cassini
			Cylindrical_Equal_Area
			Eckert4
			Eckert6
			Equidistant_Cylindrical
			Gnomonic
			Lambert_Conformal_Conic
			Mercator
			Miller_Cylindrical
			Mollweide
			Neys
			NZMG
			Oblique_Mercator
			Orthographic
			Polar_Stereo
			Polyconic

Entity Name	XML Tag Nesting	Data Type	Domain
			Sinusoidal Stereographic Transverse_Cylindrical_Equal_Area Transverse_Mercator Van_der_Grinten
Zone	Reference System Info/ projection Parameters/ zone	integer	[-60,-1] U [1,60]
Standard Parallel	Reference System Info/ projection Parameters/ standard Parallel	Decimal Latitude	0 to 2 decimal numbers of range: [-90.0,+90.0]
Longitude Of Central Meridian	Reference System Info/ projection Parameters/ longitude Of Central Meridian	Decimal Longitude	range: [-180.0, +180.0]
Latitude Of Projection Origin	Reference System Info/ projection Parameters/ latitude Of Projection Origin	Decimal Latitude	range: [-90.0,+90.0]
False Easting	Reference System Info/ projection Parameters/ false Easting	Non-Negative Decimal	[0.0, …), decimal is guaranteed at least 18 digits
False Northing	Reference System Info/ projection Parameters/ false Northing	Non-Negative Decimal	[0.0, …), decimal is guaranteed at least 18 digits
False Easting Northing Units	Reference System Info/ projection Parameters/ false Easing Northing Units	Unit Of Measure	string

Entity Name	XML Tag Nesting	Data Type	Domain
Scale Factor at Equator	Reference System Info/ projection Parameters/ scale Factor At Equator	Positive Decimal	[0.0, …)
Height of Prospective Point Above Surface	Reference System Info/ projection Parameters/ height Of Prospective Point Above Surface	Positive Decimal	[0.0, …)
Longitude of Projection Center	Reference System Info/ projection Parameters/ longitude Of Projection Center	Decimal Longitude	range: [-180.0, +180.0]
Latitude of Projection Center	Reference System Info/ projection Parameters/ latitude Of Projection Center	Decimal Latitude	range: [-90.0,+90.0]
Scale Factor at Center Line	Reference System Info/ projection Parameters/ scale Factor At Center Line	Positive Decimal	[0.0, …)
Straight Vertical Longitude from Pole	Reference System Info/ projection Parameters/ straight Vertical Longitude From Pole	Decimal Longitude	range: [-180.0, +180.0]
Scale Factor at Projection Origin	Reference System Info/ projection Parameters/ scale Factor At Projection Origin	Positive Decimal	[0.0, …)

Entity Name	XML Tag Nesting	Data Type	Domain
Oblique Line Azimuth Parameter	Reference System Info/ projection Parameters/ oblique Line Azimuth Parameter	Oblique Line Azimuth	AzimuthAngle, azimuthMeasurePointLongitude
Oblique Line Point Parameter	Reference System Info/ projection Parameters/ oblique Line Point Parameter	Oblique Line Point	obliqueLineLatitude, obliqueLineLongitude
Semi-Major Axis	Reference System Info/ Ellipsoid Parameters/ semi Major Axis	Positive Decimal	[0.0, …)
Axis Units	Reference System Info/ Ellipsoid Parameters/ axis Units	Unit Of Measure	String
Spatial Extent			
Horizontal Datum	Reference System Info/ datum/ Identifier/ code	Non-Null String	NAD83 – North American 1983
			WGS72 – World Geodetic System 1972
			WGS84 – World Geodetic System 1984
Number of Dimensions	Spatial Representation Info/ number Of Dimensions	Positive Integer	[0,1,2,…)
Resolution per Spatial Dimension	Spatial Representation Info/ Dimension/ resolution/ value	Decimal	(0.0, 1.0e18) Guaranteed 18 digits with optional ‘.’, or leading signs, '+/-'.
			[0,1,2,...,2^16-1]

Entity Name	XML Tag Nesting	Data Type	Domain
Size per Dimension	Spatial Representation Info/ Dimension/ dimension Size	nonnegative integer	
Corner Points	Spatial Representation Info/ corner Points/ Point/ coordinates	Coordinates	1 to 4 nodes of pointPropertyType decimal degrees or meters
West Bounding Longitude	Data Identification/ extent/ geographic Element/ west Bound Longitude	Approximate Longitude	[-180.00, 180.00], maximum 2 fractional digits
East Bounding Longitude	Data Identification/ extent/ geographic Element/ east Bound Longitude	Approximate Longitude	[-180.00, 180.00], maximum 2 fractional digits
South Bounding Latitude	Data Identification/ extent/ geographic Element/ south Bound Latitude	Approximate Latitude	[-90.00, 90.00], maximum 2 fractional digits
North Bounding Latitude	Data Identification/ extent/ geographic Element/ north Bound Latitude	Approximate Latitude	[-90.00, 90.00] , maximum 2 fractional digits
Bag Metadata Extension			
Tracking List ID	Data Quality/ Lineage/ process Step/ tracking Id	Positive Integer	Short (2byte) integer
Vertical Uncertainty Type	Data Identification/ vertical	Character String	Unknown = 0,

Entity Name	XML Tag Nesting	Data Type	Domain
	Uncertainty Type		
			Raw_Std_Dev = 1, CUBE_Std_Dev = 2, Product_Uncert = 3, Historical_Std_Dev = 4
Depth Correction Type	Data Identification/ depth Correction Type	Character String	Unknown = 0,
			Raw_Std_Dev = 1, CUBE_Std_Dev = 2, Product_Uncert = 3, Historical_Std_Dev = 4

Table 2.3: Elevation dataset attributes.

Entity Name	Data Type	Domain
elevation	Float 32[][]	(FLT_MIN, FLT_MAX)
Minimum Elevation Value	Float 32	(FLT_MIN, FLT_MAX)
Maximum Elevation Value	Float 32	(FLT_MIN, FLT_MAX)

Table 2.4: Uncertainty dataset attributes.

Entity Name	Data Type	Domain
uncertainty	Float 32[][]	(FLT_MIN, FLT_MAX)
Minimum Uncertainty Value	Float 32	(FLT_MIN, FLT_MAX)
Maximum Uncertainty Value	Float 32	(FLT_MIN, FLT_MAX)

Table 2.5: Tracking list dataset attributes.

Entity Name	Data Type	Domain
Tracking List Item	Bag Tracking List Item	N/A
Tracking List Length	Unsigned Integer 32	0, 2^32-1]

Table 2.6: Definition of contents of the BAG tracking list item.

Entity Name	Data Type	Domain
row	Unsigned Integer 32	location of the node of the BAG that was modified
col		

Entity Name	Data Type	Domain
	Unsigned Integer 32	location of the node of the BAG that was modified
depth	Float 32	original depth before this change
uncertainty	Float 32	original uncertainty before this change
track_code	Char	reason code indicating why the modification was made
list_series	Unsigned Integer 16	index number indicating the item in the metadata that describes the modifications

Table 2.7: Nominal elevation dataset attributes.

Entity Name	Data Type	Domain
nominal_elevation	Float 32[][]	(FLT_MIN, FLT_MAX)
Minimum Nominal Elevation Value	Float 32	(FLT_MIN, FLT_MAX)
Maximum Nominal Elevation Value	Float 32	(FLT_MIN, FLT_MAX)

Table 2.8: Node extension dataset attributes.

Entity Name	Data Type	Domain
hyp_strength	Float 32[][]	(FLT_MIN, FLT_MAX)
max_hyp_strength	Float 32	(FLT_MIN, FLT_MAX)
min_hyp_strength	Float 32	(FLT_MIN, FLT_MAX)
num_hypotheses	Unsigned Integer 32[][]	(UINT32_MIN, UINT32_MAX)
max_num_hypotheses	Unsigned Integer 32	(UINT32_MIN, UINT32_MAX)
min_num_hypotheses	Unsigned Integer 32	(UINT32_MIN, UINT32_MAX)

Table 2.9: Elevation solution extension dataset attributes.

Entity Name	Data Type	Domain
shoal_elevation	Float 32[][]	(FLT_MIN, FLT_MAX)
max_shoal_elevation	Float 32	(FLT_MIN, FLT_MAX)
min_shoal_elevation	Float 32	(FLT_MIN, FLT_MAX)
stddev	Float 32[][]	(FLT_MIN, FLT_MAX)
max_stddev	Float 32	(FLT_MIN, FLT_MAX)
min_stddev	Float 32	(FLT_MIN, FLT_MAX)
num_soundings	Unsigned Integer 32[][]	(UINT32_MIN, UINT32_MAX)
max_num_soundings	Unsigned Integer 32	(UINT32_MIN, UINT32_MAX)
min_num_soundings	Unsigned Integer 32	

Entity Name	Data Type	Domain
		(UINT32_MIN, UINT32_MAX)

Table 2.10: Georef_metadata extension dataset attributes.

Entity Name	Data Type	Domain
<LAYER_NAME_1>	HDF5 group	HDF5 group whose name corresponds to the BAG layer this georeferenced metadata layer provides metadata for.
...	...	...
<LAYER_NAME_N>	HDF5 group	HDF5 group whose name corresponds to the BAG layer this georeferenced metadata layer provides metadata for.

Each HDF5 group in `georef_metadata` must correspond to an existing BAG layer of the same name. Each `georef_metadata` group must consist of two HDF5 compound datasets: (1) "keys"; and (2) "values". The "keys" dataset is described in Table 2.11, and "values" described in Table 2.13.

Table 2.11: Required and optional elements of Georef_metadata "keys" dataset.

Entity Name	Data Type	Domain	Required
Record Definition	Array of Record Definition Items	Definition Item detailed in table 12	Yes
Metadata Profile Type	Character String	32 characters	No

Metadata Profile Type string should be "Unknown metadata profile" unless the `georef_metadata` Record Definition is of a known profile (e.g., "NOAA-OCS-2022.10"). See Appendix A for a description of known metadata profiles.

Table 2.12: Georef_metadata Record Definition Item.

Entity Name	Data Type	Domain
name	Character String	The name of the metadata record
type	Unsigned Integer 8	[UINT8_MIN, UINT8_MAX]

Table 2.13: Required elements of Georef_metadata "values" dataset.

Entity Name	Data Type	Domain
<RECORD_VALUE_1>	Float 32 or Unsigned Integer 32 or Boolean or Character String	
...	...	...
<RECORD_VALUE_N>	Float 32 or Unsigned Integer 32 or Boolean or Character String	

The raster values of a `georef_metadata` are interpreted to correspond to the entry of the "values" table containing the metadata to be associated with one or more points in raster space. The `NoData` value is 0, hence the first entry in the "values" table will always contain 0 or NULL values.

Figure 2.2 shows structure of BAG `georef_metadata` for an elevation layer encapsulated using HDF5.

```
$ h5dump -A examples/sample-data/bag_georefmetadata_layer.bag -g /
BAG_root/georef_metadata examples/sample-data/
bag_georefmetadata_layer.bag
HDF5 "examples/sample-data/bag_georefmetadata_layer.bag" {
GROUP "/BAG_root/georef_metadata" {
   GROUP "Elevation" {
      DATASET "keys" {
         DATATYPE  H5T_STD_U16LE
         DATASPACE  SIMPLE { ( 100, 100 ) / ( 100, 100 ) }
         DATA {
         (0,0): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1,
         (0,20): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1,
         (0,40): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1,
         (0,60): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1,
         (0,80): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1,
         (1,0): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1,
         (1,20): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1,
         (1,40): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1,
         (1,60): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1,
         (1,80): 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1,
         ...
         (5,0): 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,
         (5,20): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (5,40): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (5,60): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (5,80): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (6,0): 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0,
         (6,20): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (6,40): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
0, 0, 0, 0,
         (6,60): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (6,80): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         ...
         (99,0): 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (99,20): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (99,40): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (99,60): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,
         (99,80): 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0
         }
      }
      DATASET "values" {
         DATATYPE  H5T_COMPOUND {
            H5T_STD_U8LE "significant_features";
            H5T_STD_U8LE "feature_least_depth";
            H5T_IEEE_F32LE "feature_size";
            H5T_IEEE_F32LE "feature_size_var";
            H5T_STD_U8LE "coverage";
            H5T_STD_U8LE "bathy_coverage";
            H5T_IEEE_F32LE "horizontal_uncert_fixed";
            H5T_IEEE_F32LE "horizontal_uncert_var";
            H5T_STRING {
               STRSIZE H5T_VARIABLE;
               STRPAD H5T_STR_NULLTERM;
               CSET H5T_CSET_ASCII;
               CTYPE H5T_C_S1;
            } "survey_date_start";
            H5T_STRING {
               STRSIZE H5T_VARIABLE;
               STRPAD H5T_STR_NULLTERM;
               CSET H5T_CSET_ASCII;
               CTYPE H5T_C_S1;
            } "survey_date_end";
            H5T_STRING {
               STRSIZE H5T_VARIABLE;
               STRPAD H5T_STR_NULLTERM;
               CSET H5T_CSET_ASCII;
               CTYPE H5T_C_S1;
            } "source_institution";
            H5T_STRING {
               STRSIZE H5T_VARIABLE;
               STRPAD H5T_STR_NULLTERM;
               CSET H5T_CSET_ASCII;
               CTYPE H5T_C_S1;
            } "source_survey_id";
            H5T_STD_U32LE "source_survey_index";
            H5T_STRING {
```

```
            STRSIZE H5T_VARIABLE;
            STRPAD H5T_STR_NULLTERM;
            CSET H5T_CSET_ASCII;
            CTYPE H5T_C_S1;
         } "license_name";
         H5T_STRING {
            STRSIZE H5T_VARIABLE;
            STRPAD H5T_STR_NULLTERM;
            CSET H5T_CSET_ASCII;
            CTYPE H5T_C_S1;
         } "license_url";
      }
      DATASPACE  SIMPLE { ( 3 ) / ( 65535 ) }
      DATA {
      (0): {
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            NULL,
            NULL,
            NULL,
            NULL,
            0,
            NULL,
            NULL
         },
      (1): {
            0,
            1,
            1234.57,
            765.432,
            1,
            0,
            9.87,
            1.23,
            "2019-04-01 00:00:00.0Z",
            "2019-04-01 12:00:00.0Z",
            "NOAA",
            "CD71EB77-5812-4735-B728-0DC1AE2A2F3B",
            42,
            "Creative Commons Zero Public Domain Dedication
(CC0)",
            "https://creativecommons.org/publicdomain/zero/
1.0/"
         },
      (2): {
            1,
            0,
            987.6,
```

```
                6.789,
                0,
                1,
                12345.7,
                89,
                "2019-04-02 00:00:00.0Z",
                "2019-04-02 12:00:00.0Z",
                "NOAA",
                "15B46F99-1D94-4669-92D8-AA86F533B097",
                23,
                "Open Data Commons Public Domain Dedication and
Licence (PDDL)",
                "http://opendatacommons.org/licenses/pddl/1.0/"
            }
        }
      }
   }
}
}
```

Figure 2.2: Structure of BAG Georef_metadata for an elevation layer encapsulated using HDF5. Note, key datatable is truncated in several places using ellipses ("...").

Next: Axiomatic Definitions

Axiomatic Definitions

Purpose

A number of components of the BAG file structure are fixed by default. This chapter defines these components, which are used implicitly and axiomatically throughout the file structure.

Coordinate System Orientation and Geo-Referencing

All valid BAGs shall be represented with a right-handed Cartesian coordinate system. This system shall have the x-axis oriented towards positive eastings (for projected grids), or east (for geographic grids), and y-axis oriented towards positive northings (for projected grids), or north (for geographic grids). These definitions imply that the z-axis for the sounding data is positive away from the center of mass of the earth (i.e., is positive up), rather than the usual hydrographic convention of positive down (i.e., deeper depths are larger numbers and negative depths are above datum). User-level code is free to make this reflection if required but must write the data using the positive-up convention. To make this distinction clear, the term “elevation” is used for the sounding component of the BAG, rather than “depth”.

The uncertainty component of the BAG shall have the same coordinate system as the elevation component, with the exception that the z-axis is unipolar, and therefore the concept of direction of positive increase is irrelevant.

The geo-referencing for a BAG shall be node-based, referenced from the southwestern-most node in a grid. Each sample in a grid represents the value in the grid at a point location at the coordinate specified, rather than an estimate over any area with respect to the coordinate. The reference position included in the metadata shall be given in the coordinates used for the grid and shall contain sufficient digits of precision to locate the grid with accuracy no worse than a millimeter on the surface of the ellipsoid of rotation of the chosen horizontal datum.

The grid data in a BAG (either elevation or uncertainty, and any other surfaces that may be added in time) shall be organized in row-major order from west to east, and south to north in the file. The first sample of the grid is the node at the southwest corner of the grid with location as specified by the geo-referencing parameters, the second is one grid resolution unit to the east of that position and at the same northing or latitude, and the third is two grid resolution units to the east and at the same northing or latitude. For *C* columns in the grid, the (*C*+1)th sample in the grid is located one grid resolution unit to the north, but on the same easting, or longitude, as the first sample in the grid.

If a datum transformation parameter group is provided in the metadata, the interpretation shall be EPSG 9606 (position vector).

Units

The units used in all measurements are SI metric units, both in the data representation and in the metadata. Vertical measurements shall be in meters; for projected grids, the horizontal units shall be meters, and for geographic grids, the horizontal units shall be signed decimal degrees. For geographic grids, positive latitude shall be north of the equator, and positive longitude shall be degrees east of Greenwich, measured with respect to the ellipsoid of rotation associated with the horizontal datum declared in the metadata. User-level code shall ensure that geographic coordinates are appropriately mapped into the range [-180.0, 180.0]º for longitude, and [-90, 90]º for latitude.

The units of uncertainty shall be as defined by the metadata associated with the BAG. If the uncertainty can be interpreted as a variance, standard deviation, or confidence interval on the elevation data, it shall be expressed as either meters or meters squared, as appropriate for the interpretation.

Units in the metadata shall follow the units used in the grids natively. The metadata shall contain sufficient information in the geo-referencing section to allow this distinction to be determined before the user-level code has to interpret any data in the grid.

Time shall be represented in seconds UTC with respect to the standard UNIX epoch of zero seconds, 1970-01-01/00:00:00.

Resolution and Precision of Data

All data in the BAG grid, metadata and tracking list shall be represented as IEEE-754 [7] floating point numbers without rounding or limitation of precision. Data that is fundamentally integer in nature (e.g., counts of elements) may be represented in integer format for compactness. Grid data shall have single precision (32-bit) representation; metadata shall have at least single precision representation but may be more if required. All software attempting to manipulate BAGs shall at least preserve the precision of the input data.

All positioning and geo-referencing data within a BAG shall have at least millimeter resolution. All elevation information shall have at least millimeter resolution, although this should not be taken to mean that the fundamental precision of the data is better than the associated uncertainty measurement. All times shall have at least millisecond resolution.

Next: BAG Architecture Review Board

BAG Architecture Review Board

Intent of the ARB

The BAG Architecture Review Board (ARB) is intended to act as a vendor-neutral central clearing house for all activities associated with the BAG format, API, and access library. The scope of the ARB is to include:

- Organization of new releases of the library.
- Review of Request for Engineering (RFE) on the API or format, including new API calls, extensions to existing parts of the format, or addition of new HDF groups.
- Resolution of Request for Fix (RFF) on the library to address any bugs reported.
- Direction of the future development of the format, API and/or library.
- Administration of the project's website, Git revision control system, documentation and archives.

Composition and Administration

The ARB shall consist of at least three people, and preferably no more than five. The initial membership of the ARB shall be nominated from within the developer group responsible for the initial development of the library and FSD. New members may be co-opted from time to time as required by a simple majority of the current members of the ARB. Membership of the ARB shall be limited to active developers of the BAG source code library. New members may be nominated by the general public at any time by sending e-mail to the development list `navsurf_dev@ccom.unh.edu`.

RFEs and RFFs shall be accepted from any source when sent by e-mail to `navsurf_dev@ccom.unh.edu` as detailed in BAG Format Extensions. The ARB shall review the information in a timely manner and accept or reject the request based on a simple majority of the members. The results of the review shall be communicated to the donator on a best-effort basis and may be incorporated into intermediate or full releases of the BAG library at any point thereafter.

The ARB shall conduct its business primarily by e-mail but may from time to time sponsor physical meetings as required by current or pending RFEs and RFFs, or any other business as may be required. Meetings of this type shall be advertised on the project's website at least two weeks in advance of the meeting date, and by e-mail to `navsurf_general@ccom.unh.edu`. Whenever possible, the ARB shall schedule any meetings to take advantage of extant events (e.g., common-attendance hydrographic conferences) to minimize costs associated with ONS work.

Next: Revision Control and Code Availability

Revision Control and Code Availability

Baselines of the BAG source code library, external libraries, and BAG documentation are revision controlled using a Git repository hosted by the Center for Coastal and Ocean Mapping/Joint Hydrographic Center (CCOM/JHC) at the University of New Hampshire (UNH) through GitHub. Each release is referenced by a Git release tag that includes a numeric value corresponding to the release version number. Write access to these archives is granted on a case-by-case basis to active developers. Requests for write access to the Git archives can be made via email to the development list (`navsurf_dev@ccom.unh.edu`). Typically, write access should not be necessary as most collaboration can be handled via forked repositories and pull requests.

The current release is available for general download from the ONSWG website. The release distribution includes all source code required to build the BAG access library and the external library dependencies. The ONSWG web site provides contact information and a description of how to request being added to one or more of the BAG email lists. In addition, the website includes a wealth of historical background information from meeting notes, publications, and presentations.

The BAG structure is taken to be as defined by the Format Specification Document (FSD) and the corresponding release of the source code. In the case of a conflict between the FSD and the source code, precedence is given to the source code as the reference structure definition.

BAG Format Extensions details the process for reporting deficiencies identified by the user community, and for requesting extensions or enhancements to the format. Feedback and contributions from the community to improve the BAG format are encouraged. Feedback can take the form of issue reports and/or provision of implementations of actual changes to either the BAG software library or the BAG documentation.

Next: BAG Format Extensions

BAG Format Extensions

List of Extensions

Table 6.1. List of Extensions to the BAG format

Extension name	Notes
A Variable Resolution Grid Extension for BAG Files	Allow for piece-wise refinement of BAG cells to a higher resolution where necessary

How to Apply for an Extension/Bug Fix

The BAG FSD, of which this chapter is a part, is intended to be a living document, evolving as the requirements for the BAG format change. Over time, it is expected that extensions to the HDF groups in the BAG will be required, and new elements of other groups might be required. This section describes how to apply for an extension or bug fix.

Nomination Process

Any requests for extension shall be considered by the BAG Architecture Review Board as a group. All communication shall be by e-mail only, using the `navsurf_dev@ccom.unh.edu` address. Originators should include details appropriate to their request as described below and be ready to answer any subsequent questions that might be required.

A ‘receipt notice’ e-mail shall be returned to the originator immediately, and a reply to the request shall be returned as quickly as possible. The decision-making process shall be as defined in Section 6.

Request for an Extension HDF Group

Requests for an additional HDF group to be added to the base structure of the BAG must be accompanied by a full description of the data structure to be encoded. The request must be accompanied by a supporting document, e.g., an academic paper, user manual with appropriate details or a URI, and by preference code to read/write the data format. If the location of the section within the BAG structure is important, a recommendation for location may also be submitted.

The submission format may be plain text, Adobe PDF, or Microsoft Word. Other formats may be supported; Please check with the BAG-ARB before sending a document that is not in one of those three formats.

Since the FSD and the BAG format are open source, it is very important that the submission must be able to be published. This includes the source code submitted in support of the request. By sending the request to the group, the submitter explicitly agrees that:

- They are the owner of any intellectual property associated with the information in the request, and/or have the appropriate authority to transfer the associated intellectual property.
- The information in the request is not covered by any restrictions (e.g., security constraints, commercial secrets) that would prevent it from being used in the Open Navigation Surface project.
- There are no limitations on the publication, dissemination or other transmission of the data structure.
- Any source code provided may be used, adapted, or otherwise transformed for use in the source code base of the Open Navigation Surface project, including re-distribution of the code through any means in which the source code is generally made available.

Any requests that do not meet these requirements will be returned to the submitter, with comments as to cause, without further consideration by the BAG-ARB. The BAG-ARB, at their discretion, may request, in writing, confirmation of any or all the above terms, or any others as may seem appropriate at the time, from the originator of the request. Provision of this confirmation shall be a mandatory condition for acceptance and adoption of the request.

Request for an Extension to an Existing Group

Extensions to an existing group, for example adding a new ellipsoid or datum definition to the metadata, or another element to an existing group-specific metadata attribute, may be submitted in the same way as for an extension HDF group, including the requirements for free distribution and source code reuse. In addition, requests must contain a strong rationale for why the addition should be adopted.

Request for a Bug Fix

Requests for a Bug Fix should include a full description of the problem, with as much information about causes and conditions as possible, including the revision of the BAG library being used, and the version of the BAG format being constructed. If available, details about the machine's system architecture, platform, or operating system would also be beneficial to the BAG-ARG for assessing the requested bug fix.

A suitable test case should also be included, if possible, that allows the problem to be exercised. Do not send binary data attachments to the e-mail list in the first instance, since this list is distributed widely. If required, a member of the BAG-ARB will request supporting data by other means.

If a fix for the problem is known, it should be included with the initial submission. Please note that the conditions on distribution, adaptation, and re-use of source code in Section 8.2 apply to any source code submitted as a potential fix.

Next: Glossary of Acronyms

Glossary of Acronyms

Table 7.1: Glossary of Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
BAG	Bathymetric Attributed Grid
BAG-ARB	Bathymetric Attributed Grid Architecture Review Board
DS	Digital Signature
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
FSD	File Specification Document
HDF5	Hierarchical Data Format, Version 5
IEEE	Institute of Electrical and Electronic Engineers, Inc.
NS	Navigation Surface [2,3]
ONS	Open Navigation Surface
ONSWG	Open Navigation Surface Working Group
PK	Public Key
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SI	International System of Units
SK	Secret Key
URI	Universal Resource Indicator
XML	eXtensible Markup Language

Next: References

References

[1] Calder, B. R., R. T. Brennan, J. D. Case, J. S. Byrne, B. Lamey, B. Gallagher, D. Fabre, R. W. Ladner, F. Moggert, M. Paton (2005). The Open Navigation Surface Project. Int. Hydro. Review, 6(2), pp.1-10.

[2] Smith, S. M. (2003). The Navigation Surface: A Multipurpose Bathymetric Database, Masters Thesis, Center for Coastal and Ocean Mapping & Joint Hydrographic Center, University of New Hampshire.

[3] Smith, S. M., L. Alexander, & A. A. Armstrong. (2002). The Navigation Surface: A New Database Approach to Creating Multiple Products from High-Density Surveys, Int. Hydro. Review, Vol. 3, No. 2, pp. 12-26.

[4] Raymond, E. A., 2000. The Cathedral and the Bazaar. (XHTML; note that other versions, including PostScript and XML are available from the same location.).

[5] Calder, B.R., and the Open Navigation Surface Working Group (2004). Digital Security Scheme for the Bathymetric Attributed Grid (BAG), Version 1.1, February 2004. (online).

[6] Schneier, B. Applied Cryptography, 2ed. Wiley, 1995.

[7] IEEE Standard Committee 754 (1985). IEEE Standard for Floating Point Binary Arithmetic, IEEE Std. 754-1985. IEEE Inc., 345 East 47th St., New York, NY 10017, USA.

Next: Appendices

Appendices

Appendix A: Georeferenced Metadata Profile Definitions

NOAA Office of Coast Survey Metadata profile, October 2022 (NOAA-OCS-2022.10)

Table 9.1: Definition of NOAA Office of Coast Survey Metadata profile, October 2022 (NOAA-OCS-2022.10)

Column Name	Column Type	Note
significant_features	Boolean	See S-101 significant features detected.
feature_least_depth	Boolean	See S-101 least depth of detected feature measured.
feature_size	Float	See S-101 feature size (1)
feature_size_var	Float	See further discussion (1)
coverage	Boolean	See S-101 full seafloor coverage achieved
bathy_coverage	Boolean	See further discussion (2)
horizontal_uncert_fixed	Float	See S-101 horizontal position uncertainty fixed
horizontal_uncert_var	Float	See S-101 horizontal position uncertainty variable factor
survey_date_start	String	See S-101 Survey date start

Column Name	Column Type	Note
survey_date_end	String	See S-101 Survey date end
source_institution	String	e.g. "NOAA Office of Coast Survey"
source_survey_id	String	e.g. "H99999"
source_survey_index	UInt32	A value of 0 indicates the index is uninitialized or unused.
license_name	String	e.g. "CC0 1.0"
license_url	String	A URL or DOI (ideally in URL form) referencing the license definition, e.g. "https://creativecommons.org/publicdomain/zero/1.0/"

Further discussion

1. feature_size_var is meant to augment feature_size which corresponds to S-101 size of features detected. As noted in S-101, size of features detected is intended to be described as the smallest size in cubic meters the survey was capable of detecting. Depending on the type of survey this definition might force different depth ranges to have different values. For example, a survey vessel that works at a fixed height off the seafloor could maintain a fixed feature detection size capability over a wide range of depths. A surface vessel working over those same range of depths may have a feature detection capability that varies with depth causing the detection capability to be ambiguous and potentially misrepresented. For this reason, feature_size_var is the percentage of depth that a feature of such size could be detected. When both feature_size and feature_size_var are present the greater of the two should be considered valid. The expectation is that feature_size_var will be set to zero if the feature size does not scale with depth. As with feature_size, feature_size_var should be ignored if significant_features is False.

2. When side scan is used to detect features in flat seafloor areas, surveys have coverage that does not contain direct depth measurements. In these cases, the nodes with survey coverage but without bathymetry would be set to False. A condition with coverage = True and bathy_coverage = False is a useful indicator for how to work with these nodes within our workflow. If coverage is False, bathy_coverage must also be False.