DEPARTMENT OF THE NAVY
NAVAL SEA SYSTEMS COMMAND
1333 ISAAC HULL AVE SE
WASHINGTON NAVY YARD DC 20376-0001

IN REPLY TO
5720/00DT
2011 - 0107
2/9/2011

From: Commander, Naval Sea Systems Command

To: PEO LMW (Gail Baker)

Subj: Mine Countermeasures Sensor HDF Specification Version 2.0

The attached material proposed for public release has been reviewed and is returned with the following response. Amendments may be indicated on the enclosure and reasons are attached.

Statement A: Approved for Public Release; Distribution is unlimited.

for PATRICIA K. DOLAN
Director, Office of
Corporate Communications

Technical/Sensitive Material

NAVSEA Public Affairs Office

Due Date: 24 Feb 2011 **Activity:** Program Executive Office Littoral and Mine Warfare (PMS495)

Case

Title: Mine Countermeasures Sensor HDF Specification Version 2.0 dated 12 Jan 2011

Please review the enclosed material for technical accuracy and complete the form below.
Please do not put in mail.
Call Public Affairs Office (SEA00D) for pick-up at (202) 781-4122

1. Classification

TopSec _____ Sec _____ Conf _____ Unclass _ XX __

2. Is this intended for electronic dissemination?
Yes _XX___ No ____

2b. If yes, what form of electronic dissemination will be used?

World Wide Web ___X___ Internet ___X___ Other _Navy Wire

3. How will release of this hard copy material benefit the Navy? (SECNAVINST 5510.36)

Statement A: Approved for Public Release; Distribution is unlimited.

Statement B: Distribution authorized to US Government Agencies only; Other requests must be referred to COMNAVSEA or the cognizant NAVSEA code

Statement C: Distribuation authorized to US Government Agencies and their contractors; Other requests must be referred to COMNAVSEA or the cognizant NAVSEA code.

Statement D: Distribution authorized to DOD and DOD Contractors only; Other requests must be referred to COMNAVSEA or the cognizant NAVSEA code.

Statement E: Distribution authorized to DOD Components; Other requests must be referred to COMNAVSEA or the cognizant NAVSEA code.

Once released, this document will specify the foundation structure and schema for the Mine Countermeasures (MCM) Sensor Hierarchical Data Format (HDF). This new technology (MCM Sensor HDF) makes possible the management of large complex data collections and will employ HDF Version 5. This new version includes the following upgrades: a. Versatile data model which represents very complex data objects along with a wide variety of metadata. b. Portable file format with no limit on number/size of data objects in collection. c. Software library that runs on range of computational platforms & laptops to massively parallel systems & d. Integrated performance features.

Reviewer: _____ Phone: (202) 781-3937 Code: PMS495-1
Megan A. Cramer

Recommended Distribution Statement (A-F): __A__ Date 2/2/2011 Initials mc

Comments: Sensor data format for MIW systems. Compliant with MIW CoI.

Reviewer: _____ CDR USN Phone: (202) 781-2458 Code: PMS495-B
CDR Jason K. Lopez

Recommended Distribution Statement (A-F): __A__ Date 4 FEB 2010 Initials CDR LOPEZ

Comments:

Reviewer: _____ Phone: (202) 781-3903 Code: PEO(LMW)
Ms. Gail E. Baker

Recommended Distribution Statement (A-F): __A__ Date _____ Initials _____

MCM SENSOR HDF SPECIFICATION

12 JAN 2011
Version 2.0

Prepared For

Office of Naval Research & PEO LMW

CHANGE LOG

Version	Date	Changes
1.0	16 Sep 2009	Initial Release
1.1	09 Feb 2010	1) Applied proper case naming convention across all element names. 2) Changed H5S_NAVIGATION/SoundSpeed element to H5T_IEEE_F65LE from H5T_STD_U64LE 3) Changed H5S_TIME/DaylightSavings element to H5T_STD_U8LE from H5T_STD_B8LE 4) Changed H5S_CHANNELINFO/MotionCorrected element to H5T_STD_U8LE from H5T_STD_B8LE 5) Removed /Platform/SpecVersion attribute, replaced with /Platform/SpecVersionMajor and /Platform/SpecVersionMinor attributes
2.0	12 Jan 2011	Added Frame Series Datatype Added 3D Datatype Clarified Sonar PingData structure Aligned Platform Group attributes with MIW COI Guidebook Changed most datatypes to H5T_NATIVE type Added definition for MSHDF_COMPLEXNUMBER and allowed for its use in the Sonar datatype's PingData dataset Changed prefix for custom datatypes and enums from H5S_ to MSHDF_. Added GeospatialInfo attribute to store geospatial metadata compliant with ISO 19139 for FrameSeries and 3D

TABLE OF CONTENTS

TABLES

1 INTRODUCTION

The purpose of this document is to specify the foundation structure and schema for the MCM Sensor Hierarchical Data Format (HDF). HDF is a unique technology suite that makes possible the management of extremely large and complex data collections, and this specification will employ HDF version 5. The HDF5 data model, file format, API, library, and tools are open and distributed without charge at http://www.hdfgroup.org/. The HDF5 technology suite includes:

- A versatile data model that can represent very complex data objects and a wide variety of metadata.
- A completely portable file format with no limit on the number or size of data objects in the collection.
- A software library that runs on a range of computational platforms, from laptops to massively parallel systems, and implements a high-level API with C, C++, Fortran 90, and Java interfaces.
- A rich set of integrated performance features that allow for access time and storage space optimizations.
- Tools and applications for managing, manipulating, viewing, and analyzing the data in the collection.

1.1 HDF Details

The format of an HDF5 file on disk encompasses several key ideas of the HDF4 and AIO file formats as well as addressing some shortcomings therein. The new format is more self-describing than the HDF4 format and is more uniformly applied to data objects in the file.

An HDF5 file appears to the user as a directed graph. The nodes of this graph are the higher-level HDF5 objects that are exposed by the HDF5 APIs:

1) Groups
2) Datasets
3) Named datatypes

At the lowest level, as information is actually written to the disk, an HDF5 file is made up of the following objects:

1) A superblock
2) B-tree nodes
3) Heap blocks
4) Object headers
5) Object data
6) Free space

The HDF5 library uses these low-level objects to represent the higher-level objects that are then presented to the user or to applications through the APIs. For instance, a group is an object header that contains a message that points to a local heap (for storing the links to objects in the group) and to a B-tree (which indexes the links). A dataset is an object header that contains messages that describe datatype, dataspace, layout, filters, and external files, fill value, etc with the layout message pointing to either a raw data chunk or to a B-tree that points to raw data chunks. All checksums used in the format are computed with the Jenkins' lookup3 algorithm.

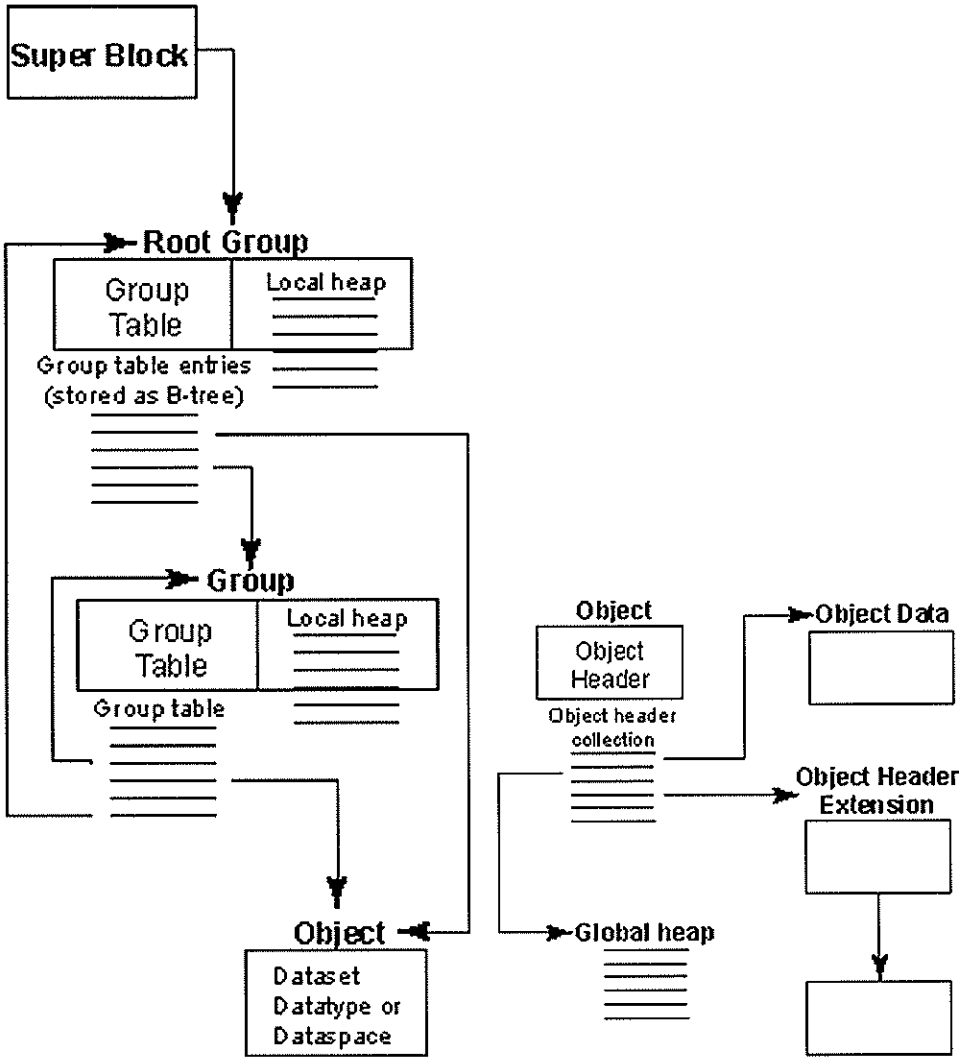


Figure 1 – Relationships among the HDF5 root group, other groups, and objects

2 ARCHITECTURE

2.1 Overview

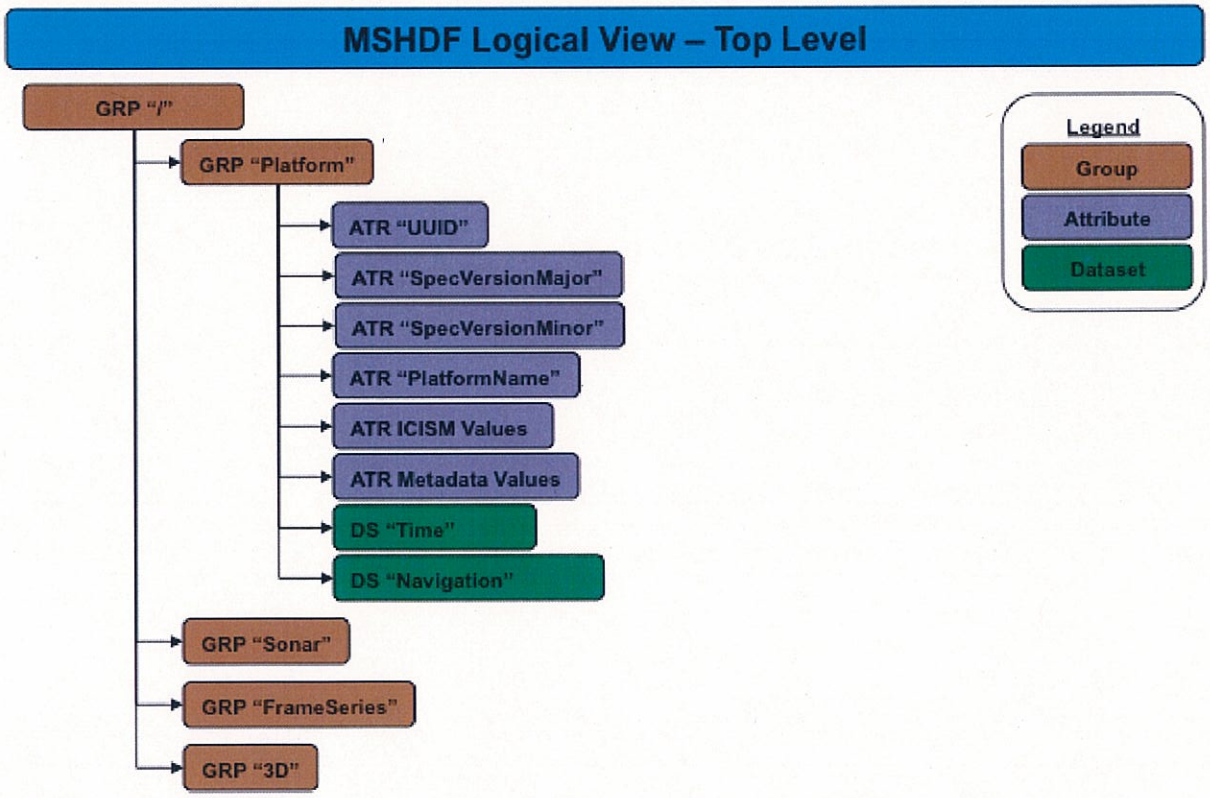
The structure of the MCM Sensor HDF foundation is intended to provide a common basis, the "base class", for storing data generated by sensor systems on platforms supporting the MCM mission profile. A number of standard data objects are defined in this base class representing the minimum set of information required to support two functions:

1. Render visual information to the screen of a post mission operator's workstation
2. Execute Computer Aided Detection / Classification (CAD/CAC) algorithms on the dataset

Additional data objects can be defined to support any specialized analysis or processing requirements for a given system/platform, and such objects should be detailed in a "derived class" specification. The flexibility of the HDF5 architecture allows this expansion beyond the base class to occur without affecting the compatibility of the data file with viewers/readers that are written to the base specification.

2.2 Base Definition

The base structure definition as summarized in the Top Level View below (Figure 2) abstracts the configuration of the system generating the data into a relational representation in HDF.



2.2.2 Platform Group

The "Platform" Group contains:

1) A unique identification number for the data set that the HDF file belongs to
2) Attributes that define the top level Platform information
3) Datasets containing the Platform's global Time and Navigation logs.

2.2.3 Datatype Groups

Each datatype defined within this specification is built from a root group structure with the name of the given datatype. As of this specification, there are three datatypes defined:

1) Sonar – Stores data from sidescan or forward looking sonars represented as scan lines
2) Frame Series – Stores series, one or more, of x-y matrices of data that can represent a diverse range of data products from mosaic tiles to independent geo-referenced images to video frames.
3) 3D – Stores three-dimensional data products in the form of either point clouds or 3D surface maps

The details for each datatype structure are defined in the following sections.

IMPORTANT NOTE

This specification defines only the base data structure that must exist under a given datatype group in order to be a valid MSHDF file. Additional datasets, attributes, and/or subgroups may be added to the datatype group or its sub-groups to support specialized processing, as long as the base structure remains intact.

2.3 Sonar Datatype

The "Sonar" group shall contain one or more "Sensor<#>" subgroup(s) representing a sensor package on the Platform that generates one or more data channel streams that can be visualized as a series of scan lines forming a two-dimensional image based on intensity at a given along-track and across-track sample point.

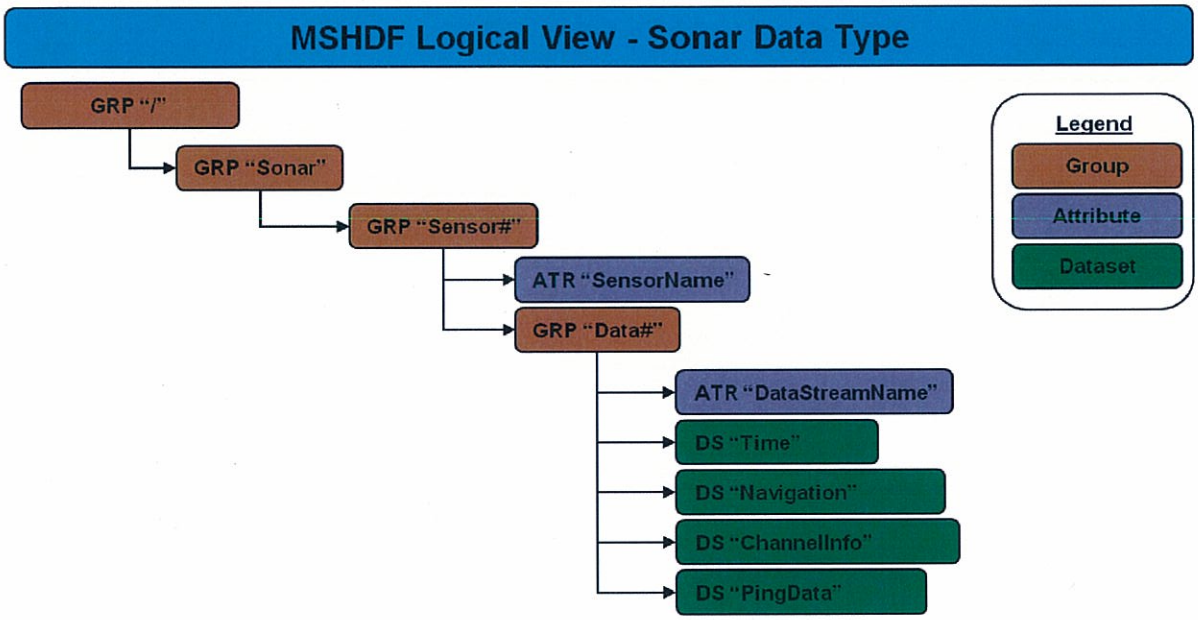


Figure 3. Sonar Data Type

2.3.1 /Sonar/Sensor# Groups

All "Sensor<#>" subgroups in the "Sonar" datatype group must contain:

1) An attribute with the sensor's identifier
2) One or more "Data<#>" subgroup(s) representing a set of data streams (1 or more) generated by the sensor that share a common Time, Navigation, and Across-track Sample Count. The # in "Data<#>" is a positive integer incrementing by one and starting at one.

2.3.2 /Sonar/Sensor#/Data# Groups

All "Data<#>" subgroups in the "Sensor<#>" group must contain:

1) An attribute with the data stream's identifier
2) "Time" and "Navigation" datasets contain the time and navigation information that correlate on integer multiple ratio (1:1, 1:2, etc.) with the PingData dataset entries.
3) A "ChannelInfo" dataset contains the information required to define the channels stored in the PingData dataset
4) A "PingData" three-dimensional dataset contains the sonar return intensity values given the ChannelInfo index (dimension 0), along-track index (dimension 1), and across-track index (dimension 2).

The key to determining the proper organization of "Data" groups given multiple data streams generated by a single "Sensor" is identifying streams that use a common time and navigation structure. The MSHDF format structure requires the along-track values stored in PingData to correlate on an integer multiple ratio to the time and navigation values in either the "Platform" group's global "Time" and "Navigation" datasets or to the "Data" group's local "Time" and "Navigation" datasets.

This provides the flexibility to minimize file size by not requiring a local "Time" and "Navigation" dataset if the sensor's data stream sample rate correlates at a determinable ratio to the Platform's time and navigation sample rate. Conversely, if the sensor's data stream is sampling at a different rate than the sampling rate of the Platform, or possibly receiving time or navigation data from a source other than the Platform, the specification supports storing the local "Time" and "Navigation" data as well.

It is the responsibility of the processing application to determine if each "Data" subgroup contains a "Time" and "Navigation" dataset. If it does, the application should correlate that "Data" group's PingData dataset to the local "Time" and "Navigation" datasets, otherwise it should correlate to the "Platform" group's global "Time" and "Navigation" datasets.

IMPORTANT NOTE

Validation shall be performed by checking the ratio between the length of the along-track dimension in the PingData dataset to the length of the single dimension in the "Time" and "Navigation" datasets. These two lengths must correlate on an integer multiple ratio such as 1:1, 1:2, 2:1, etc. In this way, if six PingData along-track lines are generated for every one Navigation/Time record, the ratio would be 6:1 and the processing application shall calculate this and use it where applicable.

The PingData dataset is a three-dimensional matrix where the cell value is the sonar ping return intensity. The matrix dimensions are:

Dimension 0 = Channel index

Dimension 1 = Along-track index – Index 0 represents the earliest scan line recorded

Dimension 2 = Across-track index – Index 0 represents the earliest return sample recorded (nadir)

In this way, data streams that share a Time, Navigation, along-track, and across-track resolution can all be rapidly accessed from the same PingData dataset. Figure 4 shows a notional PingData matrix structure with the 3rd dimension correlating to the ChannelInfo dataset records. For example, to return the ping intensity at the 50th across-track sample from the channel defined in ChannelInfo[1] at the 20th along-track scan line the matrix reference would be PingData[1, 19, 49] = Intensity Value (19 and 49 because the indexes are zero based).

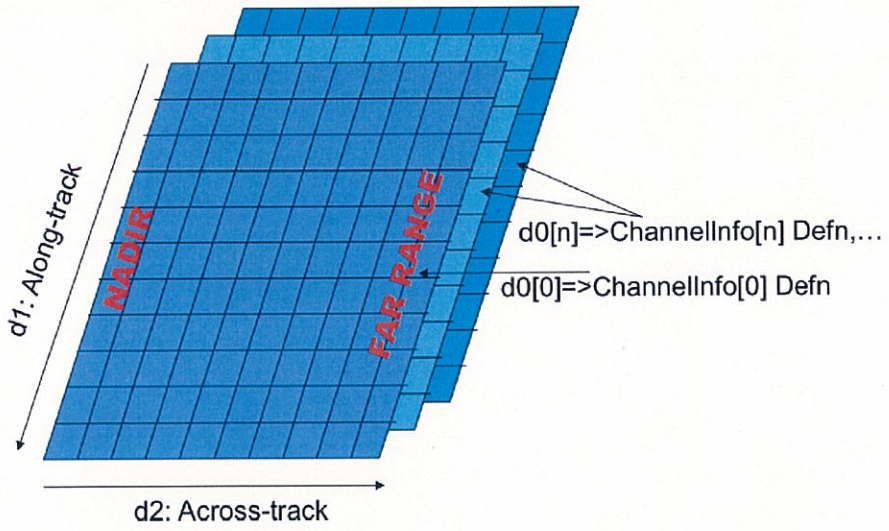


Figure 4 - PingData Structure

Along-Track Dimension: The scan line recorded at the earliest time (or recorded at the earliest along-track location) is stored at index 0, incrementing to the scan line recorded at the latest time stored at the last index.

Across-track Dimension: The across-track value of the earliest return (closest to the source / nadir) is stored at index 0, incrementing to the value of the latest return (furthest from the source). This should be used for both Port and Starboard "CHANNELTYPE"s.

For Forward Looking, SubBottom, and Bathy CHANNELTYPEs, a "starboard first, centered" convention will be maintained in keeping with the along-track ordering. The across-track value generated to the furthest starboard extent of the source sensor will be stored at index 0, incrementing up to the center value (the value closest to/at the source) stored at index (n-1)/2 (where n is the total number of across-track values), and then incrementing up to the furthest port value stored at index n-1.

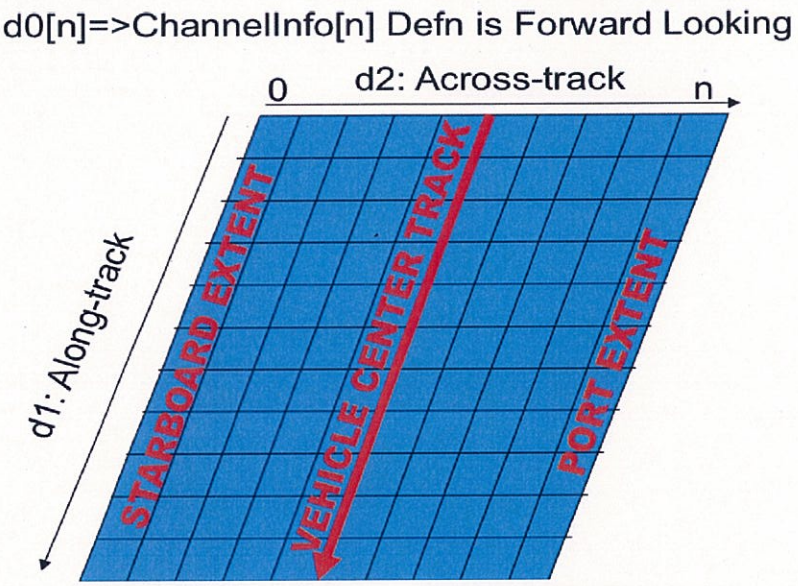


Figure 5. Forward Looking / SubBottom / Bathy Channel Layout

The relationship between the "Sensor" groups, "Data" groups, and "PingData" datasets can best be illustrated in another example, shown in Figure 6.

The SampleTech sonar payload generates four streams of data:

- High Frequency Port and Standard channels (2 streams)
- Low Frequency Port and Starboard channels (2 streams)

The two high frequency channels have the same ping rate (and so the same ping:time/nav ratio) and the same along-track and across-track sample count, but the ping rate is twice the rate that the Platform records its Time and Nav data so it records its own navigation and time records. The two low frequency channels have the same ping rate, along-track, and across-track sample count, and the ping rate is half the rate that the Platform records its Time and Nav data so it correlates to the global navigation and time records at a 1:2 ratio.

This example shows the Platform "12INUUV" recorded time and navigation datasets with 2500 samples. It has a single sensor, "SampleTech5000", in the "Sonar" datatype that contains two "Data" groups. The "Data1" group, "HighFrequency", has its own Time and Navigation datasets with 5000 samples corresponding to 5000 along-track samples in the PingData dataset dimension 0. The ChannelInfo dataset contains two entries that identify 2 channel data streams accessible in PingData's dimension 2 with a channelType of 1 (Port) and 2 (Starboard). The "Data2" group, "LowFrequency", has a similar setup, though since it correlates to the global Navigation and Time dataset it does not contain its own Navigation and Time datasets.

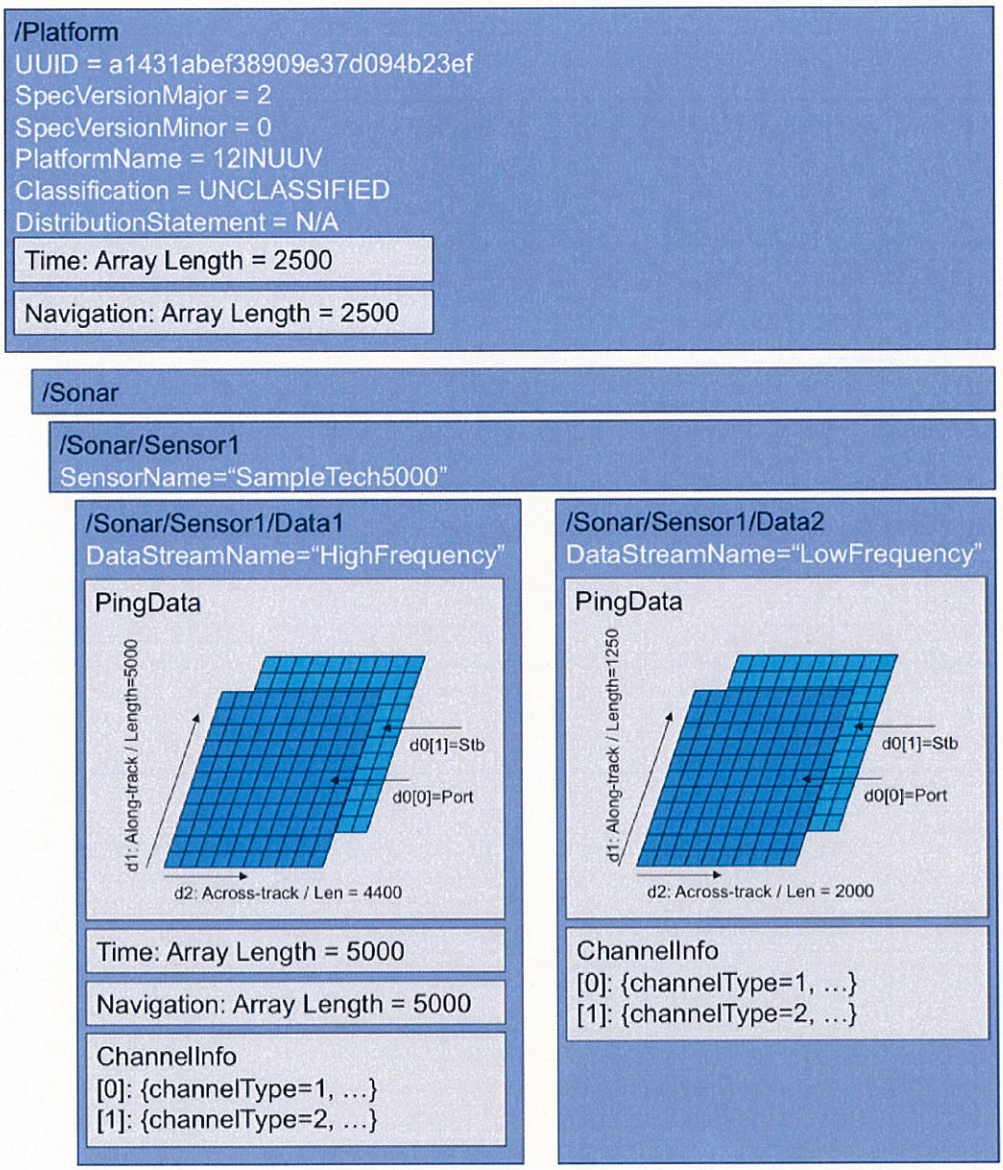


Figure 6 - Sonar Data Example

The processing application would then determine the following configuration for this file:

DataStream "HighFrequency": Correlate /Sonar/Sensor1/Data1/PingData records to /Sonar/Sensor1/Data1/Navigation and /Sonar/Sensor1/Data1/Time at a 1:1 ratio.

DataStream "LowFrequency": Correlate /Sonar/Sensor1/Data2/PingData records to /Platform/Navigation and /Platform/Time at a 1:2 ratio.

2.4 FrameSeries Datatype

The "FrameSeries" group shall contain one or more "Sensor<#>" subgroup(s) representing a sensor package on the Platform generating one or more datastreams forming one or more two-dimensional, fixed-size, geo-referenced images. This series of images may be consecutively tiled mosaic images, independently geo-referenced images, consecutive frames of video, etc.

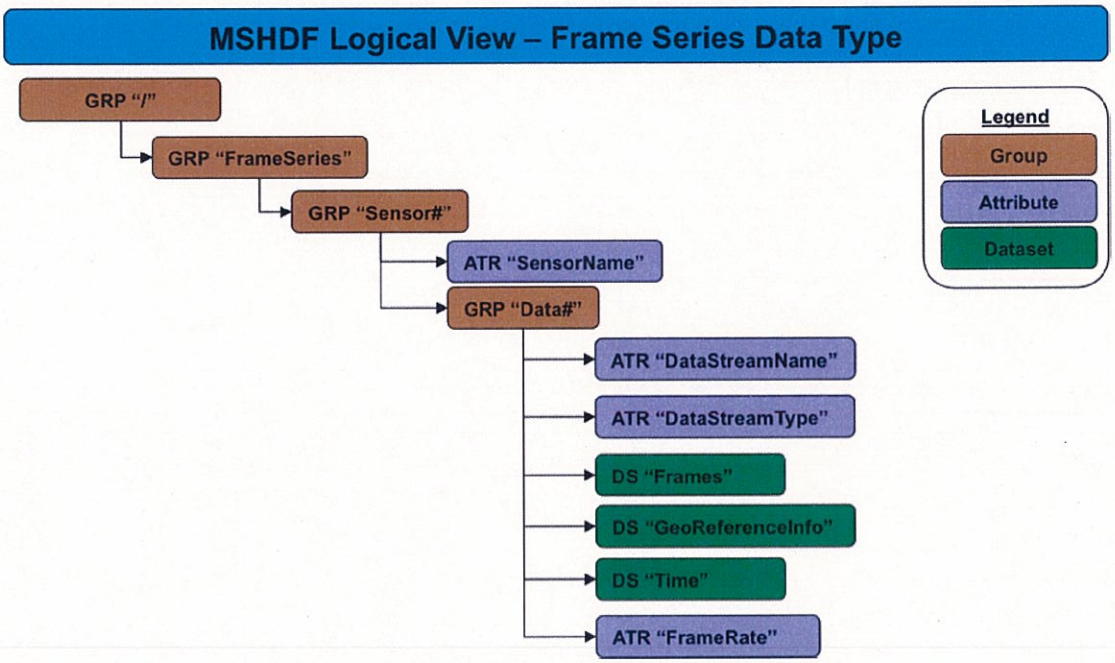


Figure 7 - Frame Series Structure

2.4.1 /FrameSeries/Sensor# Groups

All "Sensor<#>" subgroups in the "Sonar" datatype group must contain:

1) An attribute with the sensor's identifier
3) One or more "Data<#>" subgroup(s) representing a set of data streams (1 or more) generated by the sensor that shares a common pixel height and width. The # in "Data<#>" is a positive integer incrementing by one and starting at one.

2.4.2 /FrameSeries/Sensor#/Data# Groups

All "Data<#>" subgroups in the "Sensor<#>" group must contain:

1) A "DataStreamName" attribute with the data stream's identifier
2) A "DataStreamType" attribute defining the type of data stored in the series
3) A "Frames" dataset containing the collection of 2D image value matrices
4) A "GeospatialInfo" dataset that contains XML strings defining the geo-reference information for each frame in the "Frames" dataset correlated by index.

If the "DataStreamType" is VIDEO:

5) A "Time" dataset containing the timestamp at which the frame was generated.
6) A "FrameRate" attribute that defines the capture rate in frames/sec.

The Frames dataset is defined with a three-dimensional dataspace. The zero dimension (d0) is the Frame Index, dimension 1 represents the y (latitude) axis, and dimension 2 represents the x (longitude) axis.

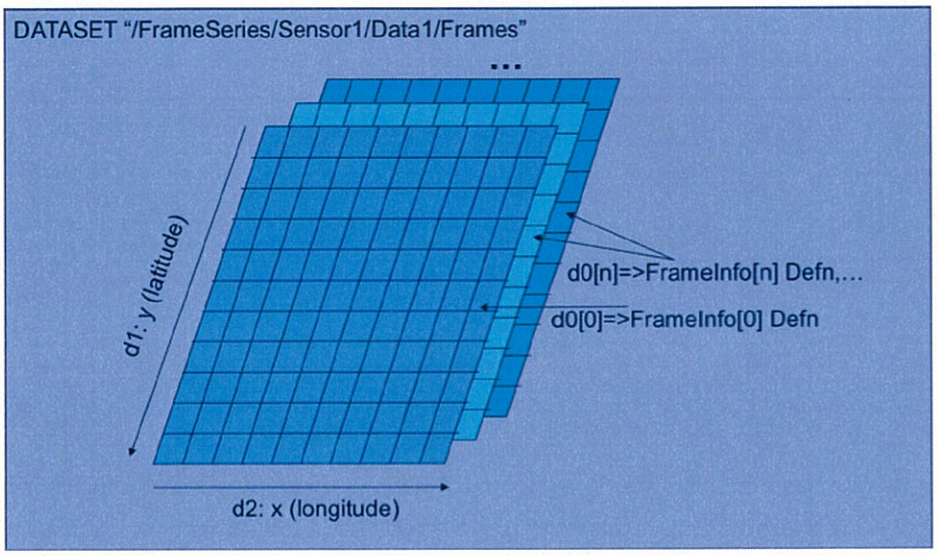


Figure 8. Frames Structure

The datatype for the Frames dataset (normally representing color) is defined by the generating system as an 8-, 16-, 24- or 32-bit integer value. The frame can then be rendered as a 2D image with 8-, 16-, or 32-(ARGB) bit color depth.

The frame construct is intended to support a myriad of 2D geo-referenced image data products. They include, but are not limited to:

- Overhead Geo-referenced Imagery
- Mosaic Tiles
- Video Frames
- Forward Looking Sonar Fixed Frames (similar to video for rendering purposes i.e. DIDSON, Blueview)
- Non-contiguous Imagery Patches
- Color Maps

Since each frame can be individually geo-referenced, it is at the discretion of the generating system to set the metadata and anchor the frames in such a way that they are:

1. Contiguous (Mosaic tiles, Stitched Imagery, etc.)
2. Non-contiguous (Color Maps, 2d Patches generated from FLS data, etc.)
3. Fixed frame with a defined frame-rate (Video, FLS Fixed Frame, etc.)

Each Data<#> group should store only one "product type" of FrameSeries data for the purpose of storage clarity. While the data is structurally separated by Sensor at a higher level, some sensors do generate multiple data product types, such as Atlas generating FLS, Color Maps, and Imagery Patches. For this example, the SensorName could be "ATLAS" with three Data groups: Data1 with DataStreamName "FLS" and DataStreamType of "VIDEO", Data2 with DataStreamName "ColorMap" and DataStreamType of "IMAGE", and Data3 with DataStreamName "IntensityPatches" and DataStreamType of "IMAGE".

2.5 3D Datatype

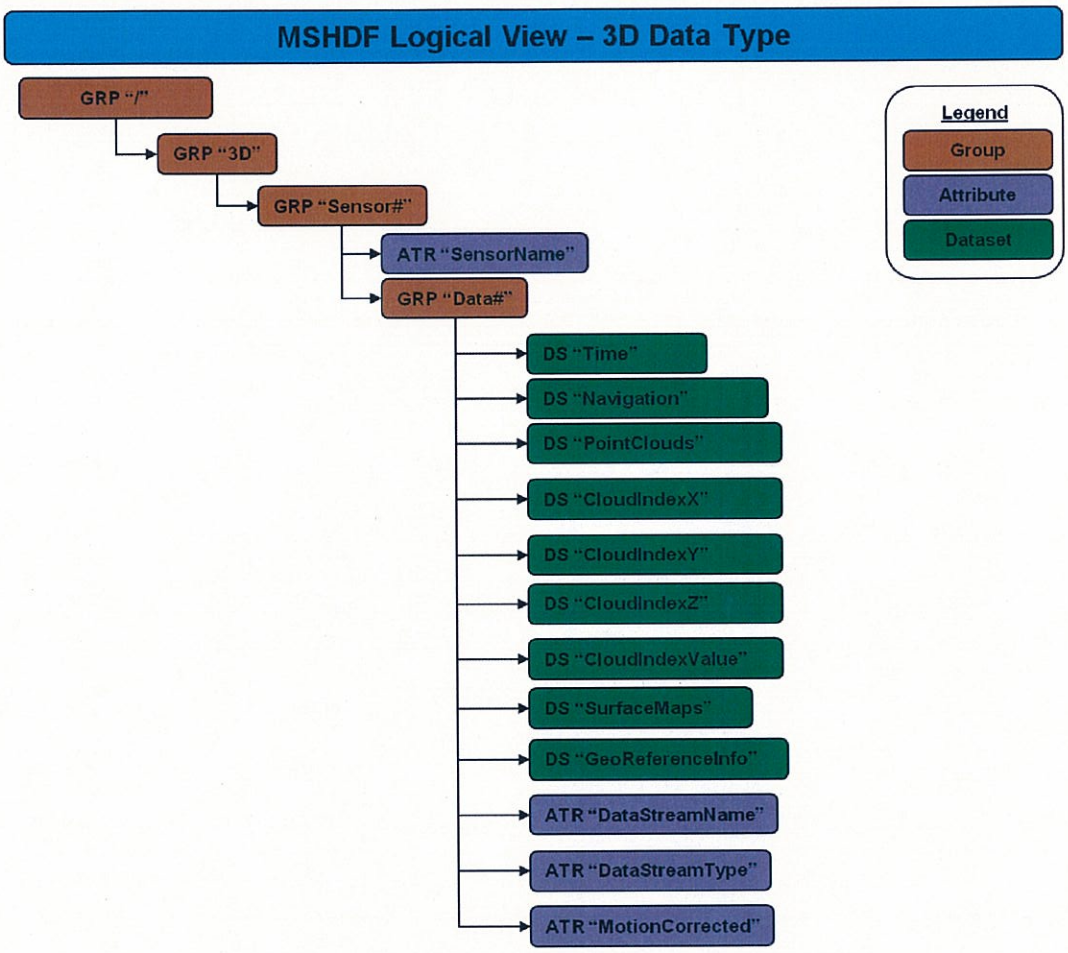


Figure 9. 3D Datatype Structure

2.5.1 /3D/Sensor#/Data# Groups

All "Data<#>" subgroups in the "Sensor<#>" group must contain:

1) A "DataStreamName" attribute with the data stream's identifier
2) "Time" and "Navigation" datasets contain the time and navigation information that correlate on integer multiple ratio (1:1, 1:2, etc.) with the PointClouds or SurfaceMaps dataset elements, if applicable.
3) A "DataStreamType" attribute with the type of 3D stored in the group: Point Cloud or Surface Map
4) A "GeospatialInfo" dataset contains XML strings defining the geo-reference metadata for the Point Cloud or Surface Maps.
5) If the DataStreamType is PointClouds, a "PointClouds" dataset that contains the point cloud vector values and four index datasets that contain sorted index references to the PointClouds dataset
6) If the DataStreamType is SurfaceMap, a "SurfaceMaps" dataset that contains the 3D surface map frames
7) A "MotionCorrected" attribute to define if the point clouds or surface maps have been corrected for sensor motion errors

DataStreamType: PointCloud

The structure of the point cloud data is designed to store the 3D data in a form requiring the least amount of processing or manipulation for both writing the original data as well as rendering with 3D software libraries. The standard process for rendering 3D point graphics with 3D libraries such as DirectX or OpenGL requires setting up the rendering device with the proper color, texture, mesh, etc., looping through and rendering that color/texture/mesh at a set of given points (3D vectors), then setting the device to the next color/texture/mesh, looping/rendering, and so on.

Given this process, the MSHDF Point Cloud is structured to store collections of "3D Points" (an x,y,z 3D vector plus a sensor value) in a dataset and provide sorted indices of the cloud's 3D points in four separate "Index" datasets. This allows the data to be written into the PointClouds dataset with no rigid order, reducing write-time overhead. After data collection, a sort algorithm populates the Index datasets with an ascending value order.

In this way, the rendering software can use any index (x, y, z, or value) to rapidly extract data in order or subsets, returning an array of 3D vectors, set the render device based on the color map and then render all the points in the subset array. Figure 10 depicts an example showing a PointClouds dataset and the corresponding CloudIndexX and CloudIndexValue datasets (CloudIndexY and CloudIndexZ not shown).

PointClouds Dataset

d0: Cloud Index

d 1: Point Index	0	1
0	5.0, 5.0, 5.0, 100	5.0, 5.0, 5.0, 100
1	2.0, 2.0, 2.0, 50	3.0, 3.0, 3.0, 70
2	3.0, 3.0, 3.0, 70	1.0, 1.0, 1.0, 10
3	1.0, 1.0, 1.0, 10	2.0, 2.0, 2.0, 50

CloudIndexX

d0: Cloud Index

d 1: Index Ref.	0	1
0	3	2
1	1	3
2	2	1
3	0	0

CloudIndexValue

d0: Cloud Index

d 1: Index Ref.	0	1
0	3	2
1	1	3
2	2	1
3	0	0

Figure 10. PointClouds Index Example

This structure would suffice if the requirement was to store only a single, global point cloud. The 3D sensors driving the MSHDF requirements generate a point cloud every time the sensor "fires", essentially taking a snapshot cloud that is in the frame of reference of the sensor's location and orientation at the time of the sampling. In this way, a series of clouds are generated, each with a potentially variable origin for the frame of reference, especially if the sensor is in motion. This is illustrated in Figure 11.

Given this, there is a requirement to store the point cloud in such a way that the each collection of points generated during a sampling event is identifiable / separable from another. This is accomplished by defining the PointClouds dataset with two dimensions, one to index the sampling event, and one to index the points.

The segmentation of point clouds based on sampling events allows for advanced processing such as navigation correction. The Time and Navigation datasets store a record correlating to each "Cloud Index" in the PointClouds dataset dimension 0, anchoring each cloud's geo-referenced origin (x=0,y=0,z=0) point.

The 3DPoint compound datatype is defined by four values: three vector distance components and the sensor measurement value. This allows for retaining the actual measured sensor value for a given point, while not requiring the rendering software to read, sort, and normalize every single point at run-time because that is already handled by the indexing.

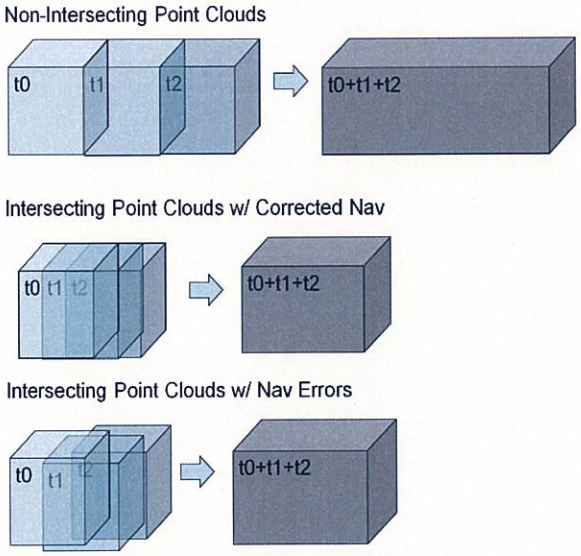


Figure 11. Point Cloud Collection Combination

DataStreamType SurfaceMap

The SurfaceMaps dataset is defined with a three-dimensional dataspace. The zero dimension (d0) is the map index, dimension 1 represents the y (latitude) axis, and dimension 2 represents the x (longitude) axis. This allows the storage of multiple surface maps in the SurfaceMaps dataset, each individually geo-referenced by index correlation to the Time and Navigation datasets.

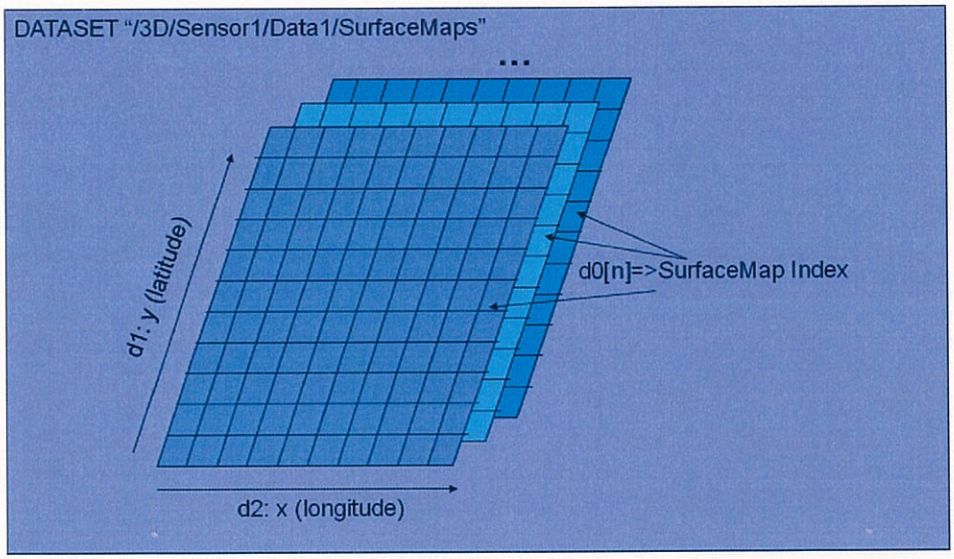


Figure 12. SurfaceMaps Structure

The SurfaceMaps datatype is defined by the generating system as integer or floating point to represent the z axis value (elevation / altitude). The 3D surface can then be rendered by faces with vertices defined by the x and y index converted to physical locations using the geo-reference anchor point and corresponding "AxisScale" attribute values, along with the stored z values.

3 SPECIFICATION REFERENCE

Note on custom datatypes:

This specification defines several custom HDF datatypes and enumerations that are written to the HDF output file. These custom datatypes and enumerations, which are not included in HDF's predefined datatype list, are prefixed with MSHDF_. Examples are MSHDF_TIME, MSHDF_NAVIGATION, MSHDF_INT128, etc. These custom datatypes, most of which are compound datatypes, are specifically defined in this document.

Notes on empty data fields:

If a particular data value is unavailable or invalid, that value should be written as 0, rather than being left as uninitialized memory or written as some other garbage value. For example, if Roll data is not available for a particular vehicle/sonar setup, then every roll value within that Navigation dataset should be written as 0. The value 0 was chosen because it seems least likely to cause problems in processing applications if there is confusion over data validity. 0 will generally either be a non-disruptive valid value (e.g. MotionCorrected, Roll) that won't inadvertently alter the data, or a clearly invalid value (e.g. Range, SoundSpeed) and thus easy to identify as something to be ignored.

3.1 Root Group

Table 1 – Root Group "/"

Object Type	Name	Datatype	Dataspace
GROUP	Platform	N/A	N/A
GROUP	Sonar	N/A	N/A
GROUP	FrameSeries	N/A	N/A
GROUP	3D	N/A	N/A
DATATYPE	MSHDF_NAVIGATION	H5T_COMPOUND	N/A
DATATYPE	MSHDF_TIME	H5T_COMPOUND	N/A
DATATYPE	MSHDF_COMPLEXNUMBER	H5T_COMPOUND	N/A

3.1.1 /Platform Group

This group contains all the global elements representing the top-level Platform parameters and data. This is because no attributes can be defined in the root group. Exactly one "Platform" group is defined per file.

3.1.2 /Sonar Group

The "Sonar" group is a datatype group storing data channel streams that can be visualized as a series of scan lines forming a two-dimensional image based on intensity at a given along-track and across-track sample point.

3.1.3 /FrameSeries Group

The "FrameSeries" group is a datatype group storing collections of two-dimensional, fixed-size geo-referenced images. This series of images may be consecutively tiled mosaic images, independently geo-referenced, consecutive frames of video, etc.

3.1.4 /3D Group

The "3D" group is a datatype group storing a collection of groups representing sensors generating 3D data products.

3.1.5 /MSHDF_NAVIGATION Compound Datatype

This datatype defines the compound structure for storing Navigation records as detailed below.

Table 2 – Compound Datatype "/MSHDF_NAVIGATION"

Name	Datatype
Yaw	H5T_NATIVE_DOUBLE
Pitch	H5T_NATIVE_DOUBLE
Roll	H5T_NATIVE_DOUBLE
Depth	H5T_NATIVE_DOUBLE
Altitude	H5T_NATIVE_DOUBLE
SoundSpeed	H5T_NATIVE_DOUBLE
Latitude	H5T_NATIVE_DOUBLE
Longitude	H5T_NATIVE_DOUBLE

3.1.5.1 Yaw

This compound attribute stores a 64-bit floating point value that represents the yaw of the Platform in degrees.

3.1.5.2 Pitch

This compound attribute stores a 64-bit floating point value that represents the pitch of the Platform in degrees.

3.1.5.3 Roll

This compound attribute stores a 64-bit floating point value that represents the roll of the Platform in degrees.

3.1.5.4 Depth

This compound attribute stores a 64-bit floating point value that represents the depth of the Platform in meters.

3.1.5.5 Altitude

This compound attribute stores a 64-bit floating point value that represents the altitude of the Platform above the sea floor in meters.

3.1.5.6 SoundSpeed

This compound attribute stores a 64-bit floating point value that represents the sound speed in the current water column in meters per second.

3.1.5.7 Latitude

This compound attribute stores a 64-bit floating point value that represents the latitude of the Platform's positions in decimal degrees.

3.1.5.8 Longitude

This compound attribute stores a 64-bit floating point value that represents the longitude of the Platform's positions in decimal degrees.

3.1.6 /MSHDF_TIME Compound Datatype

This datatype defines the compound structure for storing Time records as detailed below.

IMPORTANT NOTE

When defining the MSHDF_TIME datatype in C/C++ for reading/writing from/to a MSHDF file. C/C++ structs by default will pad each element to 4 bytes. The DaylightSavings element is one 1 byte, but C/C++ will allocate 4 bytes of memory space by default, making the total structure size 32 bytes. The MSHDF_TIME datatype should only occupy 29 bytes. In Visual Studio, you must use the #pragma pack(push, 1) directive before the struct definition, and #pragma pack(pop) directive after, to ensure that C/C++ only allocates 1 byte for the DaylightSavings element. Other compilers may implement this differently. When writing to or reading from the MSHDF file you should use the MSHDF_TIME datatype definition in the file and not use a type defined on the fly. This will ensure that a proper 29-byte MSHDF_TIME datatype is used to read the Time dataset, thus avoiding data corruption.

Table 3 – Compound Datatype "/MSHDF_TIME"

Name	Datatype
Year	H5T_NATIVE_UINT16
Month	H5T_NATIVE_UINT8
Day	H5T_NATIVE_UINT8
Hour	H5T_NATIVE_UINT8
Minute	H5T_NATIVE_UINT8
DecimalSec	H5T_NATIVE_FLOAT
DaylightSavings	H5T_NATIVE_UINT8
LocalOffset	H5T_NATIVE_FLOAT

3.1.6.1 Year

This compound attribute stores an unsigned 16-bit integer value that represents the year this record was created in four digits, i.e. 2009

3.1.6.2 Month

This compound attribute stores an unsigned 8-bit integer value that represents the month this record was created.

3.1.6.3 Day

This compound attribute stores an unsigned 8-bit integer value that represents the day of the month this record was created.

3.1.6.4 Hour

This compound attribute stores an unsigned 8-bit integer value that represents the hour component of the time this record was created.

3.1.6.5 Minute

This compound attribute stores an unsigned 8-bit integer value that represents the minute component of the time this record was created.

3.1.6.6 DecimalSec

This compound attribute stores a 32-bit floating point value that represents the second component of the time this record was created.

3.1.6.7 DaylightSavings

This compound attribute stores an 8-bit unsigned integer value. The position 0 bit represents a Boolean value that determines if daylight savings time applies to the time this record was created. So a value of

zero indicates no daylight savings is applied, any value greater than zero indicates daylight savings is applied.

3.1.6.8 LocalOffset

This compound attribute stores a 32-bit floating point value that represents the time zone hour offset. If the time was stored in UTC, this would be 0.

3.1.7 /MSHDF_COMPLEXNUMBER Compound Datatype

This datatype defines the compound structure for storing complex numbers as detailed below.

Table 4 – Compound Datatype "/MSHDF_COMPLEXNUMBER"

Name	Datatype
A	Any NATIVE INT or FLOAT
B	Any NATIVE INT or FLOAT

3.1.7.1 A

This compound attribute stores any Native integer or floating point value that represents the real component, A, of the complex number A + iB where i = (-1)^-1/2.

3.1.7.2 B

This compound attribute stores any Native integer or floating point value that represents the imaginary component, B, of the complex number A + iB where i = (-1)^-1/2.

3.2 Platform Group

Table 5 – Group "/Platform"

Object Type	Name	Datatype	Dataspace
ATTRIBUTE	UUID	MSHDF_INT128	SIMPLE 1/1
ATTRIBUTE	SpecVersionMajor	H5T_NATIVE_UINT16	SIMPLE 1/1
ATTRIBUTE	SpecVersionMinor	H5T_NATIVE_UINT16	SIMPLE 1/1
ATTRIBUTE	PlatformName	H5T_C_S1	SCALAR
ATTRIBUTE	Classification	H5T_C_S1	SCALAR
ATTRIBUTE	DistributionStatement	H5T_C_S1	SCALAR
ATTRIBUTE	ICISM_Security.ownerProducer	H5T_C_S1	SCALAR
ATTRIBUTE	ICISM_Security.releasableTo	H5T_C_S1	SCALAR
ATTRIBUTE	Metadata.contact	H5T_C_S1	SCALAR
ATTRIBUTE	Metadata.dateStamp	H5T_C_S1	SCALAR
DATASET	Time	MSHDF_TIME	SIMPLE .../UNLIMITED
DATASET	Navigation	MSHDF_NAVIGATION	SIMPLE .../UNLIMITED

3.2.1 UUID Attribute

This attribute stores the universal unique identifier for a set of HDF files representing a single mission run. A UUID is a 128-bit integer value that can be generated and read by most standard programming languages. If a single UUV mission run generates 100 HDF files, each file would have the same UUID. That UUID would then be used as a reference for that particular mission set of HDF files in any external databases, applications, etc.

The UUID Attribute maps to the MIW COI LDM Metadata.fileIdentifier attribute.

Since HDF5 does not officially define a 128-bit-wide integer datatype, we call this datatype MSHDF_INT128. In practice, it is a standard integer datatype that is modified with H5Tset_precision(...) to 128 bits.

3.2.2 SpecVersionMajor Attribute

This attribute stores an unsigned 16-bit integer value that contains the version major component of the MCM Sensor HDF specification that this file was generated in compliance with. i.e. a file created in compliance with this specification, Version 2.0 of the MSHDF Spec shall have SpecVersionMajor = 2.

3.2.3 SpecVersionMinor Attribute

This attribute stores an unsigned 16-bit integer value that contains the version minor component of the MCM Sensor HDF specification that this file was generated in compliance with. i.e. a file created in compliance with this specification, Version 2.0 of the MSHDF Spec shall have SpecVersionMinor = 0.

3.2.4 PlatformName Attribute

This attribute stores a string value containing the name of the Platform this file is representing. i.e. "MK 18 MOD 1", "MK 18 MOD 2", etc.

3.2.5 Classification Attribute

This attribute stores a string value that defines the level of classification of the data within the file. The string must be one of the allowed values listed in Table 6. The Classification attribute maps to the MIW COI LDM SecurityConstraints.classification and ICISM_Security.classification attributes.

Table 6 – Classification String Values

String Value
UNKNOWN
PUBLIC RELEASE
UNCLASSIFIED
CONFIDENTIAL
SECRET
TOP SECRET

3.2.6 DistributionStatement Attribute

This attribute stores a string containing the Department of Defense distribution statement that specifies the allowable distribution of this data based on its sensitivity and/or classification.

The DistributionStatement attribute maps to the MIW COI LDM SecurityConstraints.handling Description and ICISM_Security.DisseminationControls attributes.

3.2.7 ICISM_Security.ownerProducer Attribute

This attribute is in accordance with the MIW COI LDM Guidebook. It stores a string value identifying the national government or international organization that have purview over the classification marking of an information resource or portion therein. This element is always used in conjunction with the Classification element. Taken together, the two elements specify the classification category and the type of classification (US, non-US, or Joint). Within protected internal organizational spaces this element may include one or more indicators identifying information which qualifies as foreign government information for which the source(s) of the information must be concealed. Measures must be taken prior to dissemination of the information to conceal the source(s) of the foreign government information.

3.2.8 ICISM_Security.releasableTo Attribute

This attribute is in accordance with the MIW COI LDM Guidebook. It stores a string value identifying the country or countries and/or international organization(s) to which classified information may be released based on the determination of an originator in accordance with established foreign disclosure procedures. This element is used in conjunction with the Dissemination Controls element.

3.2.9 Metadata.contact Attribute

This attribute is in accordance with the MIW COI LDM Guidebook. It stores a string value identifying the point of contact for the organization responsible for the data.

3.2.10 Metadata.dateStamp Attribute

This attribute is in accordance with the MIW COI LDM Guidebook. It stores a string value containing the date and time the data was produced. The string should conform to the ISO 8601 International Standard,

taking the format YYYY-MM-DD HH:MMZ. i.e. 2010-02-04 24:00Z. All times should be in Coordinated Universal Time (UTC).

3.2.11 /Platform/Time Dataset

This dataset stores a one-dimensional array of MSHDF_TIME compound data records that represent the timestamps in the Platform's global log correlating in a direct 1:1 to the records in the "/Platform/Navigation" dataset.

3.2.12 /Platform/Navigation Dataset

This dataset stores a one-dimensional array of MSHDF_NAVIGATION compound data records that represent the navigation data in the Platform's global log correlating in a direct 1:1 to the records in the "/Platform/Time" dataset.

3.3 Sonar Group

Table 7 – Group "/Sonar"

Object Type	Name	Datatype	Dataspace
GROUP	Sensor<#>	N/A	N/A
DATATYPE	MSHDF_CHANNELINFO	H5T_COMPOUND	N/A
DATATYPE	MSHDF_ENUM_CHANNELTYPE	H5T_ENUM	N/A

3.3.1 /Sonar/Sensor<#> Group

This group represents a sensor on the Platform generating Sonar data. One or more of these groups can be created in the "/Sonar" group, named by appending an integer value to the string "Sensor" incrementing by one and starting at 1.

Examples: "/Sonar/Sensor1", "/Sonar/Sensor2", etc.

3.3.2 /Sonar/MSHDF_CHANNELINFO Compound Datatype

This datatype defines the compound structure for storing information about a given sonar data channel as detailed below.

Table 8 – Compound Datatype "/Sonar/MSHDF_CHANNELINFO"

Name	Datatype
Range	H5T_NATIVE_FLOAT
RangeDelay	H5T_NATIVE_FLOAT
Frequency	H5T_NATIVE_DOUBLE
AlongTrackResolution	H5T_NATIVE_DOUBLE
AcrossTrackResolution	H5T_NATIVE_DOUBLE
SlantRange	H5T_NATIVE_DOUBLE
ChannelType	/Sonar/MSHDF_ENUM_CHANNELTYPE
MotionCorrected	H5T_NATIVE_UINT8

3.3.2.1 Range

This compound attribute stores a 32-bit floating point value that represents the range the sonar returns stored in this channel in meters.

3.3.2.2 RangeDelay

This compound attribute stores a 32-bit floating point value that represents the distance in meters from Platform of sonar returns that have been cropped from the sonar return values stored for this channel in the PingData dataset. i.e. if a sonar has a 30 meter range, but the first 5 meters of sonar intensity returns are cropped from the dataset by the sonar processor, then the range value would be 30.0 and the RangeDelay would be 5.0.

3.3.2.3 Frequency

This compound attribute stores a 64-bit floating point value that represents the sampling frequency in Hz of the sonar processor analog to digital conversion.

3.3.2.4 AlongTrackResolution

This compound attribute stores a 64-bit floating point value that represents the resolution of the PingData values between along track samples in meters per along-track sample (pixel).

3.3.2.5 AcrossTrackResolution

This compound attribute stores a 64-bit floating point value that represents the resolution of the PingData values between across-track samples in meters per across-track sample (pixel).

3.3.2.6 SlantRange

This compound attribute stores a 64-bit floating point value that represents the slant range of the sonar sensor in meters.

3.3.2.7 ChannelType

This compound attribute stores a MSHDF_ENUM_CHANNELTYPE enumeration value that represents the type of channel this data stream was generated by.

3.3.2.8 MotionCorrected

This compound attribute stores an 8-bit unsigned integer value. Bits 0 through 2 represent Boolean values that define if the channel's PingData values have already been corrected for pitch, roll, and yaw motion. A Boolean value of true (1) represents that the data has been corrected for the specific motion, a value of false (0) represents that no correction has been made. The default value is 0 (no correction).

Bits 0 – 2 correspond to:

Bit 0: Pitch Corrected
Bit 1: Roll Corrected
Bit 2: Yaw Corrected
Bit 3-8: Reserved

Example: 0x04 Hex = 4 Decimal = b00000100 = Yaw Corrected, Pitch and Roll Not Corrected
Example: 0x07 Hex = 7 Decimal = b00000111 = Pitch, Roll, and Yaw Corrected

3.3.3 /Sonar/MSHDF_ENUM_CHANNELTYPE Enumeration

This enumeration defines the allowable values for representing a type of sonar data channel as detailed below.

Table 9 – Enumeration "/Sonar/MSHDF_ENUM_CHANNELTYPE"

Integer	String Value
0	UNKNOWN
1	PORT
2	STARBOARD
3	SUBBOTTOM
4	BATHYMETRY
5	FORWARD LOOKING

For Forward Looking, SubBottom, and Bathy CHANNELTYPEs, a "starboard first, centered" convention will be maintained in keeping with the along-track ordering. The across-track value generated to the furthest starboard extent of the source sensor will be stored at index 0, incrementing up to the center value (the value closest to/at the source) stored at index (n-1)/2 (where n is the total number of across-track values), and then incrementing up to the furthest port value stored at index n-1.

3.4 /Sonar/Sensor<#> Group

Table 10 – Group "/Sonar/Sensor#"

Object Type	Name	Datatype	Dataspace
ATTRIBUTE	SensorName	H5T_C_S1	SCALAR
GROUP	Data#	N/A	N/A

3.4.1 /Sonar/Sensor#/SensorName Attribute

This attribute stores a string value containing the name of the sensor this group represents.

3.4.2 /Sonar/Sensor#/Data<#> Group

This group represents a collection of data stream channels that have a common navigation and time reference frame and common along-track and across-track sample count. In this way, multiple sets of channel data can be stored in a single PingData dataset (defined in the next section), reducing redundancy and increasing processing efficiency.

One or more of these groups can be created in the "/Sensor<#>" group, named by appending an integer value to the string "Data" incrementing by one and starting at 1.

Examples: "/Sonar/Sensor1/Data1", "/Sonar/Sensor1/Data2", etc.

3.5 /Sonar/Sensor#/Data<#> Group

Table 11 – Group "/Sonar/Sensor#/Data#"

Object Type	Name	Datatype	Dataspace
ATTRIBUTE	DataStreamName	H5T_C_S1	SCALAR
DATASET	Time	MSHDF_TIME	SIMPLE .../UNLIMITED
DATASET	Navigation	MSHDF_NAVIGATION	SIMPLE .../UNLIMITED
DATASET	ChannelInfo	/Sonar/MSHDF_CHANNELINFO	SIMPLE .../UNLIMITED
DATASET	PingData	Multiple Allowable Types: Any NATIVE UINT or FLOAT or MSHDF_COMPLEXNUMBER	SIMPLE {...,...,...} / {UNL, UNL, UNL}

3.5.1 DataStreamName Attribute

This attribute stores a string containing the common name that describes the collection of data stream channels stored in the PingData dataset.

3.5.2 Time Dataset

The /Sonar/Sensor<#>/Data<#>/Time dataset is a one-dimensional array storing MSHDF_TIME compound data records that represent the timestamps in this data stream's Nav/Time frame of reference correlating at a direct 1:1 ratio to the records in the "/Sonar/Sensor<#>/Data<#>/Navigation" dataset.

3.5.3 Navigation Dataset

The /Sonar/Sensor<#>/Data<#>/Navigation dataset is a one-dimensional array storing MSHDF_NAVIGATION compound data records that represent the navigation data in this data stream's Nav/Time frame of reference correlating at a direct 1:1 ratio to the records in the "/Sonar/Sensor<#>/Data<#>/Time" dataset.

3.5.4 ChannelInfo Dataset

This dataset is a one-dimensional array storing MSHDF_CHANNELINFO compound data records storing information defining each data stream channel stored in the PingData dataset. ChannelInfo[n] contains the MSHDF_CHANNELINFO that defines the data stream channel stored in PingData[y][x][n], where n is the channel index number.

3.5.5 PingData Dataset

This dataset is a three-dimensional array storing sonar intensity values in one of multiple allowable unsigned (non-negative) integer or floating point datatypes including, but not limited to, lengths of 8, 16, 32, and 64 bits. This allows for minimization of storage requirements in that a sensor only generating 8-bit data would not have to store it in 32-bit values.

IMPORTANT NOTE

Version 2.0 of this specification adds the ability to handle the storage of the intensity values as complex numbers in PingData dataset. This is accomplished by using the MSHDF_COMPLEXNUMBER compound datatype defined in Section 3.1.7.

Along-Track Dimension: The scan line recorded at the earliest time (or recorded at the earliest along-track location) is stored at index 0, incrementing to the scan line recorded at the latest time stored at the last index.

Across-track Dimension: The across-track value of the earliest return (closest to the source / nadir) is stored at index 0, incrementing to the value of the latest return (furthest from the source). This should be used for both Port and Starboard "CHANNELTYPE"s.

The processing application shall discover the datatype of the PingData elements and use it where applicable.

The array dimensions represent:

- Dimension 0 = ChannelInfo Array index
- Dimension 1 = Along-track index
- Dimension 2 = Across-track index

Example: PingData[0][99][4] contains the intensity value for the 5^{th} across-track sample from the 100^{th} along-track sonar scan line generated by the channel defined in the 0^{th} index of the ChannelInfo dataset.

3.6 FrameSeries Group

This group is a sensor type representation. It contains a collection of groups representing sensors generating FrameSeries data products. Exactly one FrameSeries group is defined per file.

Table 12 – Group "/FrameSeries"

Object Type	Name	Datatype	Dataspace
GROUP	Sensor<#>	N/A	N/A

3.6.1 /FrameSeries/Sensor<#> Group

This group represents a sensor on the Platform generating FrameSeries data. One or more of these groups can be created in the "/FrameSeries" group, named by appending an integer value to the string "Sensor" incrementing by one and starting at 1.

Examples: "/FrameSeries/Sensor1", "/FrameSeries/Sensor2", etc.

3.7 /FrameSeries/Sensor<#> Group

Table 13 – Group "/FrameSeries/Sensor#"

Object Type	Name	Datatype	Dataspace
GROUP	Data#	N/A	N/A
ATTRIBUTE	SensorName	H5T_C_S1	SCALAR

3.7.1 Data<#> Group

This group represents a collection of data stream channels that have a common frame based data product type defined in the FrameType attribute.

One or more of these Data groups can be created in the "/Sensor<#>" group, named by appending an integer value to the string "Data" incrementing by one and starting at 1.

Examples: "/FrameSeries/Sensor1/Data1", "/FrameSeries/Sensor1/Data2", etc.

3.7.2 SensorName Attribute

This attribute stores a string value containing the name of the sensor this group represents.

3.8 /FrameSeries/Sensor<#>/Data<#> Group

Table 14 – Group "/FrameSeries/Sensor#/Data#"

Object Type	Name	Datatype	Dataspace
ATTRIBUTE	DataStreamName	H5T_C_S1	SCALAR
ATTRIBUTE	DataStreamType	H5T_C_S1	SCALAR

DATASET	Frames	User Defined: H5T_NATIVE_UINT8 or H5T_NATIVE_UINT16 or H5T_NATIVE_UINT32	SIMPLE {...,...,...} / {UNL, UNL, UNL}
DATASET	GeospatialInfo	H5T_C_S1	SIMPLE .../UNLIMITED
DATASET	Time	MSHDF_TIME	SIMPLE .../UNLIMITED
ATTRIBUTE	FrameRate	H5T_NATIVE_INT32	SCALAR

3.8.1 Frames Dataset

This dataset has a three-dimensional dataspace. The datatype can be defined as an 8-, 16-, or 32-bit unsigned integer.

The dataspace dimensions represent the following:

Dim 0: A "frame" index for storing 2d data using the 2nd and 3rd dimensions and correlation to the FrameInfo and ProjectionInfo.

Dim 1: The y-axis index for a given frame that corresponds to Latitude, where each index step represents a physical scale as specified in the FrameInfo[n].YAxisScale field.

Dim 2: The x-axis index for a given frame that corresponds to Longitude, where each index step represents a physical scale as specified in the FrameInfo[n].XAxisScale field.

3.8.2 DataStreamName Attribute

This attribute stores a string containing the common name that describes the collection of data streams stored in the Frames dataset.

3.8.3 DataStreamType Attribute

This attribute stores a string value defining the type of frame data stored in the series to assist the reading application in determining how to handle/render it. This attribute can have only one of the following values:

"IMAGE"
"VIDEO"

3.8.4 GeospatialInfo Dataset

The "GeospatialInfo" dataset stores strings of XML data that defines the geo-reference metadata for each frame. The metadata strings are correlated to the frame by dataset index. Reference Appendix A for XML schema reference and samples.

3.8.5 Time Dataset

If the DataStreamType is "VIDEO", the "Time" dataset stores the timestamps correlating to the Frames by index.

3.8.6 FrameRate Attribute

If the DataStreamType is "VIDEO", this attribute stores an integer value defining the rate at which the frames were recorded in frames/sec.

3.9 3D Group

This group is a datatype representation. It contains a collection of groups representing sensors generating 3D data products. Exactly one 3D group is defined per file.

Table 15 – Group "/3D"

Object Type	Name	Datatype	Dataspace
GROUP	Sensor<#>	N/A	N/A
DATATYPE	MSHDF_3DPOINT		

3.9.1 /3D/MSHDF_3DPOINT Compound Datatype

This datatype defines the compound structure for storing information about a single 3D Point in the Cloud as detailed below.

Table 16 – Compound Datatype "/3D/MSHDF_ 3DPOINT"

Name	Datatype
X	H5T_NATIVE_FLOAT or H5T_NATIVE_DOUBLE
Y	H5T_NATIVE_FLOAT or H5T_NATIVE_DOUBLE
Z	H5T_NATIVE_FLOAT or H5T_NATIVE_DOUBLE
SensorValue	H5T_NATIVE_FLOAT or H5T_NATIVE_DOUBLE

3.9.1.1 X Field

This field stores a floating point number that defines the distance value for the X component of the 3D vector in meters.

3.9.1.2 Y Field

This field stores a floating point number that defines the distance value for the Y component of the 3D vector in meters.

3.9.1.3 Z Field

This field stores a floating point number that defines the distance value for the Z component of the 3D vector in meters.

3.9.1.4 SensorValue Field

This field stores a floating point number representing the exact value measured by the sensor at this 3D vector point.

3.9.2 Sensor<#> Group

This group represents a sensor on the Platform generating 3D data. One or more of these groups can be created in the "/3D" group, named by appending an integer value to the string "Sensor" incrementing by one and starting at 1.

Examples: "/3D/Sensor1", "/3D/Sensor2", etc.

3.10 /3D/Sensor<#> Group

Table 17 – Group "/3D/Sensor#"

Object Type	Name	Datatype	Dataspace
GROUP	Data#	N/A	N/A
ATTRIBUTE	SensorName	H5T_C_S1	SCALAR

3.10.1 Data<#> Group

This group represents a collection of data stream channels that have a common navigation and time reference frame and common 3D DataStreamType.

One or more of these Data groups can be created in the "/Sensor<#>" group, named by appending an integer value to the string "Data" incrementing by one and starting at 1.

Examples: "/3D/Sensor1/Data1", "/3D/Sensor1/Data2", etc.

3.10.2 SensorName Attribute

This attribute stores a string value containing the name of the sensor this group represents.

3.11 /3D/Sensor#/Data# Group

Table 18 – Group "/3D/Sensor#/Data#"

Object Type	Name	Datatype	Dataspace
DATASET	Time	MSHDF_TIME	SIMPLE .../UNLIMITED
DATASET	Navigation	MSHDF_NAVIGATION	SIMPLE .../UNLIMITED
DATASET	PointClouds	/3D/MSHDF_3DPOINT	SIMPLE {...,...,...} / {UNL, UNL}
DATASET	CloudIndexX	H5T_NATIVE_UINT32	SIMPLE {...,...,...} / {UNL, UNL}
DATASET	CloudIndexY	H5T_NATIVE_UINT32	SIMPLE {...,...,...} / {UNL, UNL}
DATASET	CloudIndexZ	H5T_NATIVE_UINT32	SIMPLE {...,...,...} / {UNL, UNL}
DATASET	CloudIndexValue	H5T_NATIVE_UINT32	SIMPLE {...,...,...} / {UNL, UNL}
DATASET	SurfaceMaps	H5T_NATIVE_INT32 or H5T_NATIVE_INT64 or H5T_NATIVE_FLOAT or H5T_NATIVE_DOUBLE	SIMPLE {...,...,...} / {UNL, UNL, UNL}
ATTRIBUTE	DataStreamName	H5T_C_S1	SCALAR
ATTRIBUTE	DataStreamType	H5T_C_S1	SCALAR
DATASET	GeospatialInfo	H5T_C_S1	SIMPLE .../UNLIMITED
ATTRIBUTE	MotionCorrected	H5T_NATIVE_UINT8	SIMPLE 1/1

3.11.1 Time Dataset

The /3D/Sensor<#>/Data<#>/Time dataset is a one-dimensional array storing MSHDF_TIME compound data records that represent the timestamps in this data stream's Nav/Time frame of reference. This datasets only dimension correlates to the first dimension (d0) in the PointClouds or SurfaceMaps Dataset.

3.11.2 Navigation Dataset

The /3D/Sensor<#>/Data<#>/Navigation dataset is a one-dimensional array storing MSHDF_NAVIGATION compound data records that represent the navigation data in this data stream's Nav/Time frame of reference. This dataset's only dimension correlates to the first dimension (d0) in the PointClouds or SurfaceMaps Dataset.

The corresponding geo-referenced navigation point anchors to:

- The (d1=0, d2=0) location in the SurfaceMaps dataset.
- The x=0,y=0,z=0 location in the PointClouds dataset's collection of points

IMPORTANT NOTE

This local Navigation dataset should contain the telemetry information (especially orientation) relating to the sensor itself, not the vehicle. This is especially important for sensors generating 3D cloud data relative to the exact orientation of the sensor head.

3.11.3 PointClouds Dataset

This dataset has a two-dimensional dataspace and a datatype of 3DPOINT.

The dataspace dimensions represent the following:

Dim 0: The single sensor event that generated the points. This index correlates to the Time and Navigation dataset d0.

Dim 1: The index of the points in this particular cloud event in no definitive order.

3.11.4 CloudIndexX Dataset

This dataset stores an ascending order index of the 3D points in the PointClouds datasets sorted by the X component. It has a two-dimensional dataspace and a datatype of unsigned integer.

The dataspace dimensions represent the following:

Dim 0: Correlation index to the PointClouds dim0

Dim 1: The ascending order index of references to locations in the PointClouds dataset.

Refer to Section 2.5.1 for an illustrated example.

3.11.5 CloudIndexY Dataset

This dataset stores an ascending order index of the 3D points in the PointClouds datasets sorted by the Y component. It has a two-dimensional dataspace and a datatype of unsigned integer.

The dataspace dimensions represent the following:

Dim 0: Correlation index to the PointClouds dim0

Dim 1: The ascending order index of references to locations in the PointClouds dataset.

Refer to Section 2.5.1 for an illustrated example.

3.11.6 CloudIndexZ Dataset

This dataset stores an ascending order index of the 3D points in the PointClouds datasets sorted by the Z component. It has a two-dimensional dataspace and a datatype of unsigned integer.

The dataspace dimensions represent the following:

Dim 0: Correlation index to the PointClouds dim0

Dim 1: The ascending order index of references to locations in the PointClouds dataset.

Refer to Section 2.5.1 for an illustrated example.

3.11.7 CloudIndexValue Dataset

This dataset stores an ascending order index of the 3D points in the PointClouds datasets sorted by the point's sensor value. It has a two-dimensional dataspace and a datatype of unsigned integer.

The dataspace dimensions represent the following:

Dim 0: Correlation index to the PointClouds dim0

Dim 1: The ascending order index of references to locations in the PointClouds dataset.

Refer to Section 2.5.1 for an illustrated example.

3.11.8 SurfaceMaps Dataset

This dataset has a three-dimensional dataspace. The datatype can be defined as one of four allowable types: H5T_NATIVE_INT, H5T_NATIVE_LONG, H5T_NATIVE_SINGLE, H5T_NATIVE_DOUBLE.

The dataspace dimensions represent the following (where N is the largest index value):

Dim 0: A "frame" index for storing 2d data using the 2^{nd} and 3^{rd} dimensions and correlation to the Time and Navigation dataset.

Dim 1: The y-axis index for a given frame that corresponds to Latitude, where each index step represents a physical scale as specified in the YAxisScale attribute. The index 0 stores the most northern value, while the index N stores the most southern value.

Dim 2: The x-axis index for a given frame that corresponds to Longitude, where each index step represents a physical scale as specified in the XAxisScale attribute. The index 0 stores the most western value, while the index N stores the most eastern value.

3.11.9 DataStreamName Attribute

This attribute stores a string containing the common name that describes the collection of data streams stored in the PointClouds or SurfaceMaps dataset.

3.11.10 DataStreamType Attribute

This attribute stores an enumeration defining whether this DataStream is storing a collection of Point Clouds or Surface Maps.

3.11.11 GeospatialInfo Dataset

The "GeospatialInfo" dataset stores strings of XML data that define the geo-reference metadata for either each Point Cloud or each Surface Map individually. The metadata strings are correlated to the Point Cloud or Surface Map by index. Reference Appendix A for XML schema reference and samples.

3.11.12 MotionCorrected

This compound attribute stores an 8-bit unsigned integer (char) value. Bits 0 through 2 represent Boolean values that define if the point clouds stored in the PointClouds dataset have already been corrected for sensor motion. A Boolean value of true (1) represents that the data has been corrected for the specific motion, a value of false (0) represents that no correction has been made. The default value is 0 (no correction).

Bits 0 – 2 correspond to:

Bit 0: X-Axis Corrected
Bit 1: Y-Axis Corrected
Bit 2: Z-Axis Corrected
Bit 3-8: Reserved

Example: 0x04 Hex = 4 Decimal = b00000100 = Z Corrected, X and Y Not Corrected
Example: 0x07 Hex = 7 Decimal = b00000111 = X, Y, and Z Corrected

Appendix A – Geospatial Metadata Reference

ISO 19139 GMD Namespace

The Geospatial Metadata namespace is the standard metadata xml namespace for geographic data description. The GMD schema is based on the ISO 19139 Schema implementation of the ISO 19115 metadata standard. The schema is available at http://www.isotc211.org/2005/ or http://schemas.opengis.net/iso/19139/20070417/.

The ISO-19139 schema is quite complex. The examples below illustrate xml strings for all three supported systems (2d Geodetic, 2d UTM, 3d Geodetic) validated against gmd.xsd.

The most efficient way to implement the standard is by translating the xsd into strongly typed software classes.

The allowed Reference System Info codes are defined in Table 19.

Table 19. ISO 19139 Reference System Codes

Code	Description
WGS84G	Geodetic (Latitude / Longitude)
326zz	UTM Northern Hemisphere (zz is Zone)
327zz	UTM Southern Hemisphere (zz is Zone)

Geospatial Metadata XML for Geodetic Projection System

```
<?xml version="1.0" encoding="UTF-8"?>
<MD_Metadata
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.isotc211.org/2005/gmd"
  xsi:schemaLocation="http://www.isotc211.org/2005/gmd http://www.isotc211.org/2005/gmd/gmd.xsd"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:gco="http://www.isotc211.org/2005/gco">
  <contact gco:nilReason="inapplicable"/>
  <dateStamp gco:nilReason="inapplicable"/>
  <spatialRepresentationInfo>
    <MD_Georectified>
      <numberOfDimensions><gco:Integer>2</gco:Integer></numberOfDimensions>
      <axisDimensionProperties>
        <MD_Dimension>
          <dimensionName><MD_DimensionNameTypeCode
codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml" codeListValue="row"/></dimensionName>
          <dimensionSize><gco:Integer>1000</gco:Integer></dimensionSize>
          <resolution><gco:Measure uom="degree">0.0000300</gco:Measure></resolution>
        </MD_Dimension>
      </axisDimensionProperties>
      <axisDimensionProperties>
        <MD_Dimension>
          <dimensionName><MD_DimensionNameTypeCode
codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml" codeListValue="column"/></dimensionName>
          <dimensionSize><gco:Integer>1000</gco:Integer></dimensionSize>
          <resolution><gco:Measure uom="degree">0.0000300</gco:Measure></resolution>
        </MD_Dimension>
      </axisDimensionProperties>
      <cellGeometry><MD_CellGeometryCode codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml"
codeListValue="point"/></cellGeometry>
      <transformationParameterAvailability><gco:Boolean>false</gco:Boolean></transformationParameterAvailability>
      <checkPointAvailability><gco:Boolean>false</gco:Boolean></checkPointAvailability>
      <cornerPoints>
        <gml:Point gml:id="botleft-topright">
```

```
            <gml:coordinates decimal="." cs="," ts=" ">
              145.5998559999999884,15.1617859999999993
              145.6604690000000062,15.2298079353903688
            </gml:coordinates>
          </gml:Point>
        </cornerPoints>
        <pointInPixel><MD_PixelOrientationCode>center</MD_PixelOrientationCode></pointInPixel>
      </MD_Georectified>
    </spatialRepresentationInfo>
    <referenceSystemInfo>
      <MD_ReferenceSystem>
        <referenceSystemIdentifier>
          <RS_Identifier>
            <code><gco:CharacterString>WGS84G</gco:CharacterString></code>                <codeSpace>
<gco:CharacterString>http://www.isotc211.org/2005/resources/crs/gmxCrs.xml</gco:CharacterString>
</codeSpace>
          </RS_Identifier>
        </referenceSystemIdentifier>
      </MD_ReferenceSystem>
    </referenceSystemInfo>
    <identificationInfo gco:nilReason="inapplicable"/>
</MD_Metadata>
```

Geospatial Metadata XML for UTM Projection System

```
<?xml version="1.0" encoding="UTF-8"?>
<MD_Metadata
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.isotc211.org/2005/gmd"
   xsi:schemaLocation="http://www.isotc211.org/2005/gmd http://www.isotc211.org/2005/gmd/gmd.xsd"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:gco="http://www.isotc211.org/2005/gco">
   <contact gco:nilReason="inapplicable"/>
   <dateStamp gco:nilReason="inapplicable"/>
   <spatialRepresentationInfo>
      <MD_Georectified>
         <numberOfDimensions><gco:Integer>2</gco:Integer></numberOfDimensions>
         <axisDimensionProperties>
            <MD_Dimension>
               <dimensionName><MD_DimensionNameTypeCode
codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml" codeListValue="row"/></dimensionName>
               <dimensionSize><gco:Integer>1000</gco:Integer></dimensionSize>
               <resolution><gco:Measure uom="meter">3.0</gco:Measure></resolution>
            </MD_Dimension>
         </axisDimensionProperties>
         <axisDimensionProperties>
            <MD_Dimension>
               <dimensionName><MD_DimensionNameTypeCode
codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml" codeListValue="column"/></dimensionName>
               <dimensionSize><gco:Integer>1000</gco:Integer></dimensionSize>
               <resolution><gco:Measure uom="meter">3.0</gco:Measure></resolution>
            </MD_Dimension>
         </axisDimensionProperties>
         <cellGeometry><MD_CellGeometryCode codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml"
codeListValue="point"/></cellGeometry>
         <transformationParameterAvailability><gco:Boolean>false</gco:Boolean></transformationParameterAvailability>
         <checkPointAvailability><gco:Boolean>false</gco:Boolean></checkPointAvailability>
         <cornerPoints>
            <gml:Point gml:id="botleft-topright">
               <gml:coordinates decimal="." cs="," ts=" ">
                  349585.8619910000124946,1676687.7601819999981672
                  356122.8619910000124946,1684202.7601819999981672
               </gml:coordinates>
            </gml:Point>
         </cornerPoints>
         <pointInPixel><MD_PixelOrientationCode>center</MD_PixelOrientationCode></pointInPixel>
      </MD_Georectified>
   </spatialRepresentationInfo>
```

```
  <referenceSystemInfo>
    <MD_ReferenceSystem>
      <referenceSystemIdentifier>
        <RS_Identifier>
          <code><gco:CharacterString>32655</gco:CharacterString></code>
          <codeSpace>
<gco:CharacterString>http://www.isotc211.org/2005/resources/crs/gmxCrs.xml</gco:CharacterString>
</codeSpace>
        </RS_Identifier>
      </referenceSystemIdentifier>
    </MD_ReferenceSystem>
  </referenceSystemInfo>
  <identificationInfo gco:nilReason="inapplicable"/>
</MD_Metadata>
```

Geospatial Metadata XML for 3D Geographic Projection with Local System

For 3D point clouds, the dimension resolutions are in meters, and locally referenced to to a (0,0,0) center point, which is geo-rectified in the world by a Latitude/Longitude coordinate center point.

```
<?xml version="1.0" encoding="UTF-8"?>
<MD_Metadata
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.isotc211.org/2005/gmd"
  xsi:schemaLocation="http://www.isotc211.org/2005/gmd http://www.isotc211.org/2005/gmd/gmd.xsd"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:gco="http://www.isotc211.org/2005/gco">
  <contact gco:nilReason="inapplicable"/>
  <dateStamp gco:nilReason="inapplicable"/>
  <spatialRepresentationInfo>
    <MD_Georectified>
      <numberOfDimensions><gco:Integer>3</gco:Integer></numberOfDimensions>
      <axisDimensionProperties>
        <MD_Dimension>
          <dimensionName><MD_DimensionNameTypeCode
codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml" codeListValue="row"/></dimensionName>
          <dimensionSize><gco:Integer>1000</gco:Integer></dimensionSize>
          <resolution><gco:Measure uom="meter">3.0</gco:Measure></resolution>
        </MD_Dimension>
      </axisDimensionProperties>
      <axisDimensionProperties>
        <MD_Dimension>
          <dimensionName><MD_DimensionNameTypeCode
codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml" codeListValue="column"/></dimensionName>
          <dimensionSize><gco:Integer>1000</gco:Integer></dimensionSize>
          <resolution><gco:Measure uom="meter">3.0</gco:Measure></resolution>
        </MD_Dimension>
      </axisDimensionProperties>
      <axisDimensionProperties>
        <MD_Dimension>
          <dimensionName><MD_DimensionNameTypeCode
codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml" codeListValue="vertical"/></dimensionName>
          <dimensionSize><gco:Integer>1000</gco:Integer></dimensionSize>
          <resolution><gco:Measure uom="meter">3.0</gco:Measure></resolution>
        </MD_Dimension>
      </axisDimensionProperties>
      <cellGeometry>
        <MD_CellGeometryCode codeList="http://www.isotc211.org/2005/resources/codelist/gmxCodelists.xml"
codeListValue="point"/>
      </cellGeometry>
      <transformationParameterAvailability><gco:Boolean>false</gco:Boolean></transformationParameterAvailability>
      <checkPointAvailability><gco:Boolean>false</gco:Boolean></checkPointAvailability>
      <centerPoint>
        <gml:Point gml:id="centerPoint">
          <gml:coordinates decimal="." cs="," ts=" ">
            21.5000, -117.2500
          </gml:coordinates>
```

```
            </gml:Point>
          </centerPoint>
          <pointInPixel><MD_PixelOrientationCode>center</MD_PixelOrientationCode></pointInPixel>
        </MD_Georectified>
      </spatialRepresentationInfo>
      <referenceSystemInfo>
        <MD_ReferenceSystem>
          <referenceSystemIdentifier>
            <RS_Identifier>
              <code><gco:CharacterString>WGS84G</gco:CharacterString></code>
              <codeSpace>
                <gco:CharacterString>http://www.isotc211.org/2005/resources/crs/gmxCrs.xml</gco:CharacterString>
              </codeSpace>
            </RS_Identifier>
          </referenceSystemIdentifier>
        </MD_ReferenceSystem>
      </referenceSystemInfo>
      <identificationInfo gco:nilReason="inapplicable"/>
</MD_Metadata>
```