

# DSM Sonar Software

*Tim Weiß, Jens Greinert*

GEOMAR Helmholtz Centre for Ocean Research, Kiel, Germany

## *Introduction*

Post-processing of sonar data is a common task in ocean science. This includes the creation of bathymetric maps, the use of seafloor backscatter information to determine sediment types as well as the analysis of water column data, e.g. for the assessment of fish populations or the quantification of gas releases from the seafloor at methane seeps. Many commercial and free post processing tools for these tasks exist, but since there is a wide range of sonar manufactures with individual data formats not every format is supported by these applications. The DSM Sonar Software is a collection of software libraries and applications to facilitate access to sonar data of various formats and convert between different formats. The conversion of an unsupported format into a commonly supported one makes it possible to process sonar data with existing post processing tools. The target users for our converters are scientists and hydrographers. The different data format descriptions and sonar tool libraries are valuable for software developers in the field of automated sonar data analysis. The modular approach and structure of the DSM Sonar Software enables that developers pick only necessary components to include them in their own software project.

## *Project overview*

The software currently consists of 31 Visual Studio projects and can be grouped into five categories shown in Figure 1 (pg. 3):

- External libraries contain open source code from third party providers
- General libraries contain general tools and sonar tool libraries
- Sonar format libraries contain decoders and encoders for different sonar formats
- Sonar format conversion libraries convert one sonar format into another
- Applications include project files for executable applications (.exe)

Many of the project files are so called test projects. These projects contain unit tests for the specified library. Currently more than 430 unit tests are implemented to ensure a working code base and also provide example code on how to use the library and as documentation of its functionality.

When sonar data sets are used for scientific publications the ability to create reproducible data sets is very important. The DSM Sonar Software libraries support individual versioning of each library to comply with this demand. The versioning process can be a part of the build process and includes the current GIT commit count.

This makes sure the state of a library can be recreated and data can be reprocessed if necessary. The code is entirely written in C/C++. The development of the software was conducted on Windows using Microsoft Visual Studio. However, major parts of the code base are platform independent and an adaptation of the source code to Linux platforms is planned for the near future.

In the following sections, the categories shown in Figure 1 are shortly described. A more extensive documentation can be found in the respective folder of the DSM Sonar Software on GEOMARs GIT server (<https://git.geomar.de/open-source/dsm-sonar-software/>).

### General Libraries

The Tools library contains functionality for common tasks. For example, it offers date time structures, conversion of different time formats, byte conversion and filesystem tools as well as logging capabilities. The Sonar library contains classes and methods which focus on sonar specific tasks. An implementation of a single beam model defines e.g. certain parameters and angles (Figure 2, pg.3). Ray tracing strategies can be applied to this model and sound velocity profiles are supported. Hence it is possible to exactly localize acoustic echoes from single and multi-beam echo sounders on the ocean floor.

### Sonar Format Libraries

Sonar manufacturers tend to use their own binary data formats. The sonar format libraries have implemented several formats to enable direct access to sonar data. Analysis and post processing tools can be developed much faster without the time consuming task of implementing codecs. The Sonar Format Libraries support the following formats:

Format	Type	Vendor	Implemented by	Decoder	Encoder
<b>XSE</b>	Bathymetry	ELAC (Wärtsila)	Tim Weiss	x	
<b>WCI</b>	Watercolumn	ELAC (Wärtsila)	Tim Weiss	x	
<b>GSF</b>	Bathymetry	Leidos	S.Ferguson/D.Chayes	x	x
<b>GWC</b>	Watercolumn	QPS	Tim Weiss	x	x
<b>83P</b>	Bathymetry	Imagenex	Tim Weiss	x	
<b>RAW</b>	Single Beam Ek15/60/80	Kongsberg	Tim Weiss	x	

### Sonar Format Conversion Libraries

The main task when converting from one sonar format to another is to map the data fields from a source format to the appropriate fields in the target format. Further it involves the adaptation of different data representations, e.g. different units or coordinate systems. Sometimes complete data representations differ from each other. For example, a 3D-point on the ocean floor can be represented using a travel time value of an acoustic wave and beam and pitch angles. In another format the point is represented as a relative point with respect to the transducer with across track, along track and depth components (see Figure 2, pg. 3). In such a case more advanced conversion strategies are required.

The sonar format conversion libraries provide three different converters:

- XSE to GSF conversion
- 83P to GSF conversion
- WCI to GWC conversion

### Applications

The applications are a series of command line tools, which use the above mentioned libraries. This could be a simple example to illustrate the usage of a library or a real application, e.g. format converter. The format converters contain some convenience functions, e.g. convert all files of a directory, use different sound velocity speeds or concatenate output files.

### Source Code

The source code is available on GEOMARs GIT server.

GIT Repository: <https://git.geomar.de/open-source/dsm-sonar-software/>

The code of the DSM Sonar Software is published as open source under the LGPL 2.1 license. More detailed information and a copy of the license can be found in the respective folder on GEOMARs GIT server.

**Acknowledgement**

The authors would like to thank Dr. Christian Zwanzig, Jörg Brockhoff, Frank Ritters and Dominic Queissner from Wärtsilä ELAC Nautik GmbH for the continuous and valuable support during the implementation of ELAC specific multibeam formats.

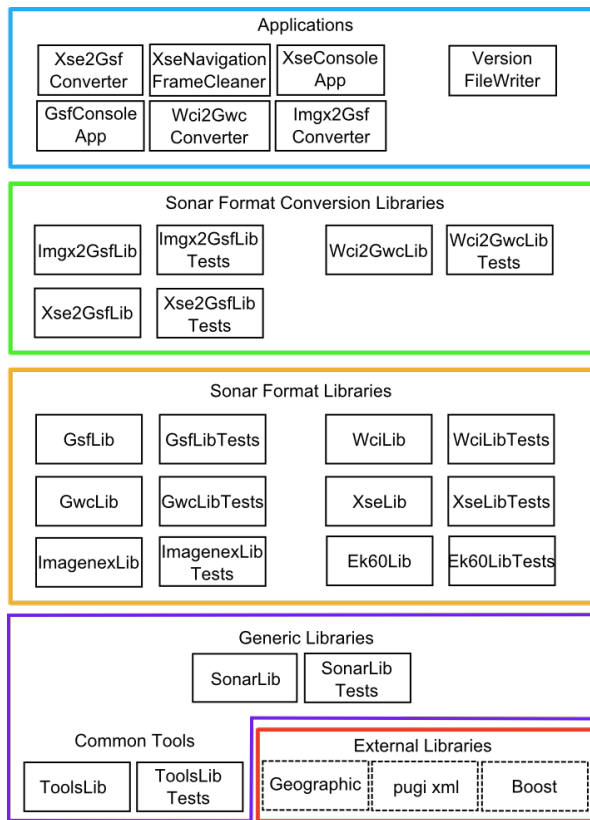


Figure 1: Project structure

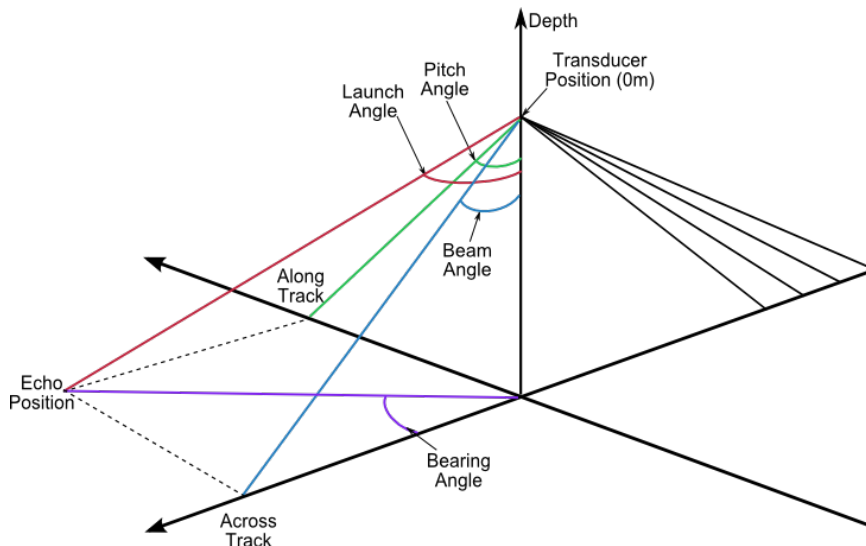


Figure 2: Single Beam Model