



GEOLOGICAL SURVEY OF CANADA

OPEN FILE xxxx

**SEGYLib - An XML-Enabled .NET C# Library used to Read, Write and Manipulate
SEGY Files**

R. C. Courtney

2017



Natural Resources
Canada

Ressources naturelles
Canada

Canada

GEOLOGICAL SURVEY OF CANADA

OPEN FILE xxx

SEGYLib – An XML-Enabled .NET C# Library used to Read, Write and Manipulate SEG Y Files

R. C. Courtney

2017

©Her Majesty the Queen in Right of Canada 2016

Available from

Geological Survey of Canada

601 Booth Street

Ottawa, Ontario K1A 0E8

R. C. Courtney

2017: SEG YLib - An XML-Enabled .NET C# Library used to Read, Write and Manipulate SEG Y Files, Geological Survey of Canada, Open File xxxx.

Open files are products that have not gone through the GSC formal publication process.

Contents

| | |
|--------------------------------------|----|
| Abstract..... | 5 |
| Introduction | 6 |
| Implementation | 7 |
| Class Hierarchy..... | 8 |
| Release Details | 9 |
| Description of Class Library..... | 10 |
| SEGYFile..... | 10 |
| Constructors..... | 10 |
| Properties..... | 10 |
| Methods..... | 10 |
| Fields | 11 |
| Sample Usage of SEGYFile | 12 |
| SEGYFileHeader Class..... | 14 |
| Constructors..... | 14 |
| Properties..... | 14 |
| Methods..... | 16 |
| Fields | 16 |
| Sample Usage of SEGYFileHeader | 17 |
| SEGYTrace Class | 18 |
| Properties..... | 18 |
| Methods..... | 18 |
| SEGYTraceData Class..... | 19 |
| Constructors..... | 19 |
| Properties..... | 19 |
| Methods..... | 19 |
| SEGYTraceHeader Class | 20 |
| Constructors..... | 20 |

| | |
|--|----|
| Properties..... | 20 |
| Methods..... | 24 |
| SEGUtilities Class..... | 26 |
| Constructors..... | 26 |
| Methods..... | 26 |
| Extending Class for Local Variants to SEG Y Standard | 27 |
| Appendix 1 – XML Schema for SEG Ylib | 30 |

SEGYPlib V1.0 – A .NET C# Library used to Read and Write SEGYP Files

Abstract

SEGYPlib V1.0 is a Microsoft C# class library that can be used within the family of Visual Studio products to read and write SEG-Y files up to and including Revision 1 (Norris and Faichney, 2002). SEG-Y is one of the formats established by the Society of Exploration Geophysicists (SEG) to standardize the storage of single-channel and multichannel seismic data. The SEG-Y standard is in the process of revision and the library released here should be capable of extension to new revisions without a complete rewrite.

This library can be used interchangeably in the Microsoft suite of Visual Studio Tools, include Visual C#, F#, Visual Basic, and Visual C++ projects. This library can also be loaded as .NET assemblies in Windows-implementations of Matlab and Python. Both the code and the complied libraries are included in this release. It is a work in progress and this release represents a preliminary functionality for reading and writing SEGYP files.

The class library is structured to support the serialization of SEGYP contents to and from XML. Entire SEGYP files, SEG-Y File header and individual SEG-Y traces can be read and written in XML format, facilitating scanning of SEG-Y files for metadata harvesting.

Keywords: seismic data, SEG-Y, C#, Visual Basic, Matlab, Python, XML

Introduction

The GSC has been collecting digital seismic data since the early 1990's and has used and continues to use SEG-Y (Norris and Faichney, 2002) as its primary format for storing its digital seismic, sounder and sidescan data. Early efforts in the 1990's at the Geological Survey of Canada Atlantic had developed computer code written in C and C++ languages to read and write SEG-Y files up to Revision 0 (Barry et al., 1975). Although these routines can still be used, they suffer from a range of issues from a programming perspective. The older code is not object-oriented so the extension, or modification, of the code often involves awkward and substantial rewrites. Older code also relied heavily on direct pointer manipulation for memory allocation and access; it is well known that this approach often results in memory leaks and code overwrites. As program complexity increases, these problems sometimes present significant barriers to progress and stable programming.

Modern coding techniques rely on an object-oriented programming (OOP) approach where these pitfalls can be addressed. In OOP, memory allocation and deallocation are strictly controlled, abstracted from the physical memory in the system. Memory leaks are eliminated as garbage collection techniques actively dispose of unused or discarded memory allocations. With proper modelling, the code becomes much more reusable and extendable. In addition, the use of structured objects leads readily to the concept of serialization and the expression of SEG-Y data in XML format, useful for harvesting metadata for data storage and dissemination.

Current versions of Visual Studio (as of 2015) are migrating away from using C++ as a primary programming language, so it was decided to code this effort in C# using object oriented programming techniques. It was decided to update the core code libraries to handle SEG-Y formatted data up to version 1 (Norris and Faichney, 2002), laying a better-structured foundation for the eventual upgrade to Revision 2.

Implementation

The SEG-Y file structure is well documented and made available through the SEG (Norris and Faichney, 2002). The reader is strongly recommended to read this reference before proceeding. The SEG-Y standard has undergone two revisions (Barry et al., 1975; Norris and Faichney, 2002) in the last 40 years, maintaining essential file and byte-level structure compatibility between revisions. It is anticipated by the author that this compatibility will be maintained through future revisions.

A SEG-Y file comprises a sequence of byte stream blocks, the structure of each strictly defined through the standard. The byte order of the file is generally big endian, however little endian versions do exist.

File Header Section:

Block 1 : 3200 byte Textual header - Traditionally IBM EBCDIC –encoded text header information. The SEG-Y standard does not explicitly state EBCDIC, and ASCII is often encountered. This implementation supports both ASCII and EBCDIC.

Block 2 : 400 byte Binary File Header as described in the standard.

Block 2+i : Extended Textual Header for $i = 0, n$. SEG-Y Revision 1 supports extended text blocks. This implementation supports from 0 to n extended text blocks. A variable text block designation (-1) is not supported at the time.

Trace Section – sequence repeated for each encoded trace

Block j - 240 byte binary trace header as described in the standard .

Block $j+1$ - trace data as described in the standard.

A class library was written to allow structured accesses to these file contents and to also permit parts, or the whole, of the SEG-Y file to be written in XML format to aid metadata harvesting. The following section details the framework of the implementation released in this open file.

Class Hierarchy

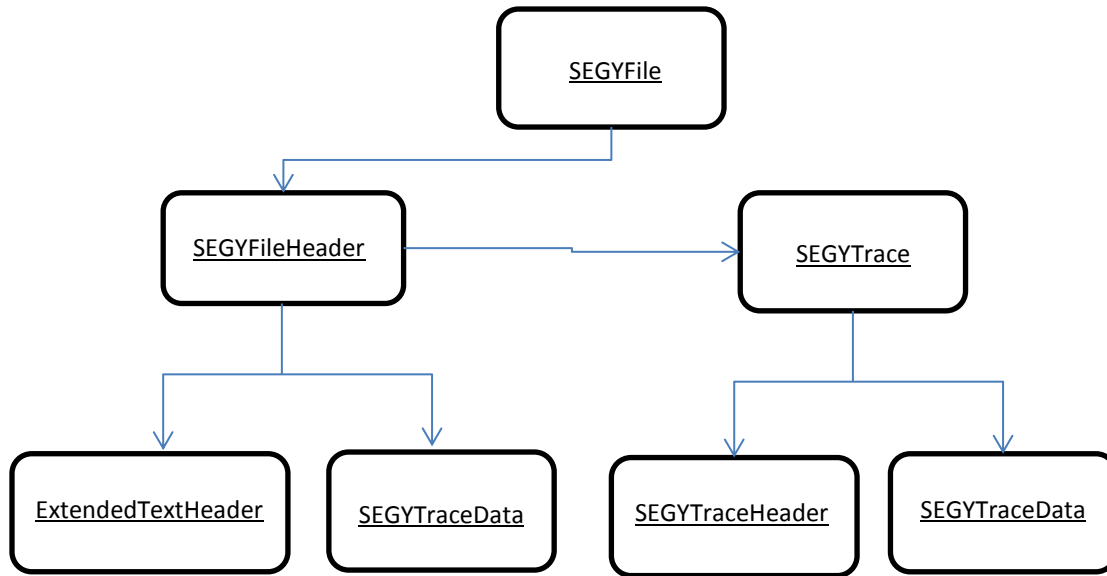


Fig. 1 Class Hierarchy

The SEG-Y has been structured into a treed class structure that strongly reflects the byte stream blocked structure upon which it was derived. The byte stream blocks from the source file are read in and stored directly in byte arrays within the class structure in their original byte order (big-endian or little endian). The SEG-Y attributes are accessed through properties that dynamically access these original byte organized block structures. This feature allows a structured pathway for the reinterpretation of SEG-Y attributes that will allow the user to accommodate local variations in the implementation of the SEG-Y standard by different vendors and organizations.

Release Details

The Visual Studio project tree containing source code and the compiled class library are included in this release as open source for unrestricted general use. The code only relies on one open source external library (URL: <http://www.codeproject.com/Articles/492449/Transform-between-IEEE-IBM-or-VAX-floating-point>) used to convert to and from IBM floating point format. This portion is subject to the Code Project Open License (CPOL) 1.02 (<http://www.codeproject.com/info/cpol10.aspx>) which is unrestrictive to any application. This release also contains an XML schema that can be used to validate XML instances of SEGYlib.

The release is in either zip or CD/DVD format and the file structure is as follows:

| Name | Path | Remarks |
|---------------------------------------|-----------------|---|
| SEGYlib.dll | SEGYLib\Release | .Net 4.5 library for SEGYlib |
| Converters.dll | SEGYLib\Release | .Net 4.5 library for IBM floating point converter |
| SEGYlib | SEGYLib | C# source tree for SEGYlib (VS2013) |
| Converters | SEGYLib | C# source tree for Converters (VS2013) |
| SEGYLib.docx | SEGYLib | MS Office 2010 version of this document |
| SEGYLib.pdf | SEGYLib | PDF version of this document |
| SEGYlib Library Reference Manual.docx | SEGYLib | MS Office 2010 version of the complete library reference manual |
| SEGYlib Library Reference Manual.pdf | SEGYLib | PDF version of the complete library reference manual |
| SEGYlib Library Reference Manual.chm | SEGYLib | Microsoft Compiled HTML Help version of the complete library reference manual |
| SEGYlib.xsd | SEGYLib | XML schema of SEGY output |
| SEGYLibInstall.zip | SEGYLib | Microsoft Install Package for SEGYLib |

Description of Class Library


A description of the main objects of the library follows:

SEGYFile





This class is primary interface to read and write SEGY rev 1 formatted files. The details of the public properties, fields and methods can be found in attached library reference guide.

The **SEGYFile** type exposes the following members.









Constructors

























| | Name | Description |
|---|--------------------------|---|
|  | SEGYFile | Initializes a new instance of the SEGYFile class |

Properties


| | Name | Description |
|---|--|---|
|  | currentTrace | last trace read from file |
|  | FileHeader | access to File Header Class |
|  | NumberOfTracesInBuffer | number of traces in Trace list |
|  | Traces | List of traces including data and trace headers |

Methods

| | Name | Description |
|---|----------------------------------|--|
|  | AddTrace | add a trace to the end of the Traces list |
|  | Close | close I/O channels |
|  | CopyAllTraces | make a deep copy of the Traces List |
|  | Equals | Determines whether the specified Object is equal to the current Object . (Inherited from Object .) |
|  | Finalize | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .) |
|  | GetHashCode | Serves as a hash function for a particular type. (Inherited from Object .) |
|  | GetType | Gets the Type of the current instance. (Inherited from Object .) |
|  | GoToStartOfTrace | position the stream reader/writer at the start of the n'th trace |

| | | |
|---|----------------------------------|--|
|  | isSEGy | test to see if input file is a SEGy file |
|  | MemberwiseClone | Creates a shallow copy of the current Object . (Inherited from Object .) |
|  | MoveFilePointerToStartOfTraces | move file pointer to the end of the file header blocks |
|  | Open | open or create a SEGy file returns 0 if unsuccessful; 1 if non zero length file ; 2 is empty file |
|  | ReadAllTraceHeaders | read all trace headers but don't load trace data |
|  | ReadAllTraces | read all trace headers including trace data |
|  | ReadFileHeader | read the file headers |
|  | ReadNextTrace | read the next trace in the file |
|  | ReadNTraces | read the next n traces in the file |
|  | ReadXML | read an SEGy file in XML format |
|  | ReadXMLFileHeader | read an SEGy file header in XML format |
|  | ReadXMLTrace | read an SEGy trace in XML format |
|  | ReindexTracePositions | re-read the file and reindex the trace locations |
|  | RemoveAllTraces | delete all trace storage |
|  | RemoveTrace | remove trace i from the Traces list |
|  | SkipNTracesOnRead | skip ntraces |
|  | ToString | Returns a string that represents the current object. (Inherited from Object .) |
|  | Write(String) | write the entire file to disk |
|  | Write(SEGYFileHeader) | write the file header to disk |
|  | Write(SEGYTrace) | write a trace to disk |
|  | Write(List(SEGYTrace)) | write the list Traces to disk |
|  | WriteXML(String) | write the file to XML |
|  | WriteXML(String, SEGyFileHeader) | write the file header to XML |
|  | WriteXML(String, SEGyTrace) | write the trace to XML |

Fields

| | Name | Description |
|---|-------------|---|
|  | isBigEndian | true for big endian file; false little endian |

Sample Usage of SEGYFile

Read an entire file

```
SEGYlib.SEGYFile sf = new SEGYlib.SEGYFile();
string inputSEGYfile = this.openFileDialog1.FileName;
sf.Open(inputFileName);
if( !sf.isSEGY() )
{
    sf.Close();
    return;
}
sf.ReadAllTraces();
```

Read only trace headers in case the file is excessive is length

```
SEGYlib.SEGYFile sf = new SEGYlib.SEGYFile();
string inputSEGYfile = this.openFileDialog1.FileName;
sf.Open(inputSEGYfile);
if( !sf.isSEGY() )
{
    sf.Close();
    return;
}
sf.ReadAllTraceHeaders();
```

Read only trace headers in case the file is excessive is length – use this form if you want to use progress bars

```
SEGYlib.SEGYFile sf = new SEGYlib.SEGYFile();
string inputSEGYfile = this.openFileDialog1.FileName;
sf.Open(inputSEGYfile);
sf.MoveFilePointerToStartOfTraces();
sf.Traces = new List<SEGYlib.SEGYTrace>();
while (sf.ReadNextTrace())
{
    SEGYlib.SEGYTrace tr = sf.currentTrace;
    tr.Data = d;
    tr.TraceData.TraceDataBuffer = null; // dump the trace data
    sf.Traces.Add(tr);
    c++;
    // put progress bar update her
}
sf.Close();
```

Create a new SEGY file from using an existing one as a template

```
SEGYlib.SEGYFile sf = new SEGYlib.SEGYFile();
inputSEGYfile = this.openFileDialog1.FileName;
sf.Open(this.openFileDialog1.FileName); // open an existing SEGY file
if( !sf.isSEGY() )
{
    sf.Close();
    return;
}

SEGYlib.SEGYFile sf2 = new SEGYlib.SEGYFile(); // create a new SEGY file
sf2.Open(outputFileName);

sf2.FileHeader = sf.FileHeader.Copy(); // copy the input trace header
sf2.Write(sf2.FileHeader); // write out the header

while ( sf.ReadNextTrace() )
{
    SEGYlib.SEGYTrace tr = sf.currentTrace;

    SEGYlib.SEGYTrace newTr = tr.Copy();

    newTr.sourcePositionX = newX; // do some operations on the traceheader
    newTr.sourcePositionY = newY;

    if ( this.checkBoxCreateMillisecondField.Checked )
    {
        newTr.TraceHeader.lagTimeBMsec = (short) millisecondsCorrectionsToShotTime[c];
        newTr.TraceHeader.timeBasis = (ushort) millisecondsCorrectionsToShotTime[c];
    }
    sf2.Write(newTr);
}

sf.Close();
sf2.Close();
```

Write out a trace in XML format

```
SEGYlib.SEGYFile sf = new SEGYlib.SEGYFile();
inputSEGYfile = this.openFileDialog1.FileName;
sf.Open(this.openFileDialog1.FileName); // open an existing SEGY file
if( !sf.isSEGY() )
{
    sf.Close();
    return;
}

sf.ReadNextTrace();
sf.WriteXML("text.xml", sf.currentTrace);
sf.Close();
```

SEGYFileHeader Class















Class used for storing and retrieving data stored in the SEGY file Header
















The **SEGYFileHeader** type exposes the following members.

Constructors















| | Name | Description |
|---|----------------|-------------|
|  | SEGYFileHeader | constructor |

Properties



| | Name | Description |
|---|--|--|
|  | amplitudeRecoveryMethod | attribute defined though segy rev 1 standard |
|  | BinaryFileHeader | access to byte block of Binary File header |
|  | binaryGainRecovered | attribute defined though segy rev 1 standard |
|  | correlatedDataTraces | attribute defined though segy rev 1 standard |
|  | dataSampleFormatCode | attribute defined though segy rev 1 standard |
|  | ensembleFold | attribute defined though segy rev 1 standard |
|  | ExtendedTextHeader | lead 3200 byte tape header plus any other extended blocks |
|  | fixedLengthTraceFlag | attribute defined though segy rev 1 standard |
|  | impulseSignalPolarity | attribute defined though segy rev 1 standard |
|  | jobIdentificationNumberz | attribute defined though segy rev 1 standard |
|  | lengthOfFileHeader | byte length of file header including extended tape header and binary file header |
|  | lineNumber | attribute defined though segy rev 1 standard |
|  | measurementSystem | attribute defined though segy rev 1 standard |
|  | numberOfAuxiliaryTracesPerEnsemble | attribute defined though segy rev 1 standard |

| | |
|--|--|
|  numberOfDataTracesPerEnsemble | attribute defined though segy rev 1 standard |
|  numberOfExtendedTextualFileHeaderRecordsFollowing | attribute defined though segy rev 1 standard |
|  numberOfSamplesPerDataTrace | attribute defined though segy rev 1 standard |
|  numberOfSamplesPerDataTraceForOriginalFieldRecording | attribute defined though segy rev 1 standard |
|  reelNumber | attribute defined though segy rev 1 standard |
|  sampleIntervalInMicroseconds | attribute defined though segy rev 1 standard |
|  sampleIntervalInMicrosecondsInOriginalFieldRecording | attribute defined though segy rev 1 standard |
|  segymFormatRevisionNumber | attribute defined though segy rev 1 standard |
|  sweepCode | attribute defined though segy rev 1 standard |
|  sweepFrequencyEnd | attribute defined though segy rev 1 standard |
|  sweepFrequencyStart | attribute defined though segy rev 1 standard |
|  sweepLength | attribute defined though segy rev 1 standard |
|  sweepTraceTaperLengthAtEnd | attribute defined though segy rev 1 standard |
|  sweepTraceTaperLengthAtStart | attribute defined though segy rev 1 standard |
|  taperType | attribute defined though segy rev 1 standard |
|  traceNumberSweepChannel | attribute defined though segy rev 1 standard |
|  traceSortingCode | attribute defined though segy rev 1 standard |
|  verticalSumCode | attribute defined though segy rev 1 standard |
|  vibratoryPolarityCode | attribute defined though segy rev 1 standard |

Methods

| | Name | Description |
|---|---|--|
|  | Copy | make a deep copy of the Header |
|  | Equals | Determines whether the specified Object is equal to the current Object . (Inherited from Object .) |
|  | Finalize | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .) |
|  | GetFileHeaderText | get a string for the extended tape header |
|  | GetFileHeaderTextByLine | get the Text header by 80 character lines |
|  | GetHashCode | Serves as a hash function for a particular type. (Inherited from Object .) |
|  | GetType | Gets the Type of the current instance. (Inherited from Object .) |
|  | isBigEndian | true for big endian and false for little endian |
|  | isFileHeaderASCII | is the file header encoded with ASCII or EBCDIC |
|  | MemberwiseClone | Creates a shallow copy of the current Object . (Inherited from Object .) |
|  | ReadFileHeader | read the file header from disk |
|  | SetFileHeader | set the Text Header by 80 character line |
|  | ToString | Returns a string that represents the current object. (Inherited from Object .) |
|  | WriteFileHeader | write the file header to disk |

Fields

| | Name | Description |
|---|---|---|
|  | isSEGyFileHeaderAscii | true if Text Header is ASCII; false if EBCDIC |
|  | positionOfStartOfDataTraces | file position of start of trace data |

Sample Usage of SEGYFileHeader

Examine parts of the file header














```
SEGYlib.SEGYFile sf = new SEGYlib.SEGYFile();
string inputSEGYfile = this.openFileDialog1.FileName;
sf.Open(inputSEGYfile);
if( !sf.isSEGY() )
{
    sf.Close();
    return;
}

string head = sf.FileHeader.GetFileHeaderText(0); // get the first header data
int code = sf.FileHeader.dataSampleFormatCode ; //read header value
```











SEGYTrace Class

SEGYTrace is used to access and set SEGY rev 1 trace data

Properties

| | Name | Description |
|---|--|---|
|  | codedTime | trace time in DDDHHHMMSSmmm |
|  | Data | signal amplitude |
|  | groupPositionXGSCDIG | GSCA implementation of group position |
|  | groupPositionYGSCDIG | GSCA implementation of group position |
|  | isBigEndian | true if big endian |
|  | isLatLon | is it a lat/lon position or projected |
|  | positionOfTraceInFile | position in bytes |
|  | sourcePositionX | source position X corrected for scaling factors |
|  | sourcePositionY | source position Y corrected for scaling factors |
|  | timeTracedRecorded | DateTime of trace instance |
|  | totalLengthOfTraceData | total number of bytes of trace data in including trace header |
|  | TraceData | access to underlying Trace Data Class |
|  | TraceHeader | access to underlying Trace Header Class |

Methods


| | Name | Description |
|---|---------------------------------|---|
|  | Copy | make a deep copy of a SEGY Trace |
|  | Equals | Determines whether the specified Object is equal to the current Object . (Inherited from Object .) |
|  | Finalize | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .) |
|  | FixMsecField | transcribe msec field in old GSC format the old GSC formatted files used the Time Basis Field 166-167 for storing msec field should use lag b or lag A field this copies 166-167 to 106-107 |
|  | GetHashCode | Serves as a hash function for a particular type. (Inherited from Object .) |
|  | GetType | Gets the Type of the current instance. (Inherited from Object .) |
|  | Intialize | initilize trace structure |
|  | MemberwiseClone | Creates a shallow copy of the current Object . (Inherited from Object .) |
|  | ToString | Returns a string that represents the current object. (Inherited from Object .) |
|  | Write | write a trace to a BinaryWriter stream |

SEGYTraceData Class




SEGYTraceData allows access to the contents of the binary trace data

The **SEGYTraceData** type exposes the following members.








Constructors

| | Name | Description |
|---|-------------------------------|--|
|  | SEGYTraceData | SEGYTraceData allows access to the contents of the binary trace data |

Properties

| | Name | Description |
|---|---------------------------------|--|
|  | Data | a double precision view of the trace data use this to read and change the contents of the trace data buffer |
|  | DataCopy | Use this if you want to change the data values as SEGYPTraceData.Data always returns values in the trace data buffer |
|  | TraceDataBuffer | access to byte[] trace data block |


Methods

| | Name | Description |
|---|---------------------------------|--|
|  | Equals | Determines whether the specified Object is equal to the current Object . (Inherited from Object .) |
|  | Finalize | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .) |
|  | GetHashCode | Serves as a hash function for a particular type. (Inherited from Object .) |
|  | GetType | Gets the Type of the current instance. (Inherited from Object .) |
|  | Initialize | Initialize the class |
|  | MemberwiseClone | Creates a shallow copy of the current Object . (Inherited from Object .) |
|  | ToString | Returns a string that represents the current object. (Inherited from Object .) |

SEGYTraceHeader Class

The **SEGYTraceHeader** type exposes the following members.

Constructors

| Name | Description |
|---|---|
|  SEGYTraceHeader | SEGYTraceHeader is used to access and change contents of the binary trace header data block |

Properties

| Name | Description |
|--|-----------------------------------|
|  aliasFilterSlopeDBOctave | refer to SEGY rev 1 documentation |
|  aliasFrequencyHz | refer to SEGY rev 1 documentation |
|  bigEndian | true if big endian |
|  coordinateUnits | refer to SEGY rev 1 documentation |
|  correlated | refer to SEGY rev 1 documentation |
|  crossLineNumber3D | refer to SEGY rev 1 documentation |
|  dataUse | refer to SEGY rev 1 documentation |
|  datumElevationAtReceiverGroup | refer to SEGY rev 1 documentation |
|  datumElevationAtSource | refer to SEGY rev 1 documentation |
|  dayOfYear | refer to SEGY rev 1 documentation |
|  delayRecordingTimeMsec | refer to SEGY rev 1 documentation |
|  deviceTraceIdentifier | refer to SEGY rev 1 documentation |
|  distanceFromCenterOfSourcePointToCenterOfGroup | refer to SEGY rev 1 documentation |
|  energySourcePointNumber | refer to SEGY rev 1 documentation |
|  ensembleNumber | refer to SEGY rev 1 documentation |



| | |
|--|------------------------------------|
|  gainTypeOfFieldInstruments | refer to SEG Y rev 1 documentation |
|  gapSize | refer to SEG Y rev 1 documentation |
|  geophoneGroupNumberofLastTraceWithinOriginalFieldRecord | refer to SEG Y rev 1 documentation |
|  geophoneGroupNumberofRollSwitchPositionOne | refer to SEG Y rev 1 documentation |
|  geophoneGroupNumberofTraceNumberOneWithinOriginalFieldRecord | refer to SEG Y rev 1 documentation |
|  groupCoordinateX | refer to SEG Y rev 1 documentation |
|  groupCoordinateY | refer to SEG Y rev 1 documentation |
|  groupStaticCorrectionMsec | refer to SEG Y rev 1 documentation |
|  highCutFrequencyHz | refer to SEG Y rev 1 documentation |
|  highCutSlopeDBOctave | refer to SEG Y rev 1 documentation |
|  hourOfDay | refer to SEG Y rev 1 documentation |
|  inLineNumber3D | refer to SEG Y rev 1 documentation |
|  instrumentEarlyOrIntialGainDB | refer to SEG Y rev 1 documentation |
|  instrumentGainConstantDB | refer to SEG Y rev 1 documentation |
|  lagTimeAMsec | refer to SEG Y rev 1 documentation |
|  lagTimeBMsec | refer to SEG Y rev 1 documentation |
|  lowCutFrequencyHz | refer to SEG Y rev 1 documentation |
|  lowCutSlopeDBOctave | refer to SEG Y rev 1 documentation |
|  minuteOfHour | refer to SEG Y rev 1 documentation |
|  muteTimeEndTimeMsec | refer to SEG Y rev 1 documentation |






| | |
|--|-----------------------------------|
|  muteTimeStartTimeMsec | refer to SEGY rev 1 documentation |
|  notchFilterSlopeDBOctave | refer to SEGY rev 1 documentation |
|  notchFrequencyHz | refer to SEGY rev 1 documentation |
|  numberOfHorizonatallySummedTracesYieldingThisTrace | refer to SEGY rev 1 documentation |
|  numberOfSamplesInTrace | refer to SEGY rev 1 documentation |
|  numberOfVerticallySummedTracesYieldingThisTrace | refer to SEGY rev 1 documentation |
|  originalFieldRecordNumber | refer to SEGY rev 1 documentation |
|  overTravel | refer to SEGY rev 1 documentation |
|  receiverGroupElevation | refer to SEGY rev 1 documentation |
|  sampleIntervalUsec | refer to SEGY rev 1 documentation |
|  scalarAppliedToShotPointNumber | refer to SEGY rev 1 documentation |
|  scalarForAllElevationsAndDepths | refer to SEGY rev 1 documentation |
|  scalarToBeAppliedToAllCoordinates | refer to SEGY rev 1 documentation |
|  scalarUsedToScaleTraceHeaderMSecTimes | refer to SEGY rev 1 documentation |
|  secondOfMinute | refer to SEGY rev 1 documentation |
|  shotpointNumber | refer to SEGY rev 1 documentation |
|  souceStaticCorrectionMsec | refer to SEGY rev 1 documentation |
|  sourceCoordinateX | refer to SEGY rev 1 documentation |
|  sourceCoordinateY | refer to SEGY rev 1 documentation |
|  sourceDepthBelowSurface | refer to SEGY rev 1 documentation |

| | |
|---|------------------------------------|
|  sourceEnergyDirectionExponent | refer to SEG Y rev 1 documentation |
|  sourceEnergyDirectionMantissa | refer to SEG Y rev 1 documentation |
|  sourceMeasurementExponent | refer to SEG Y rev 1 documentation |
|  sourceMeasurementMantissa | refer to SEG Y rev 1 documentation |
|  sourceMeasurementUnit | refer to SEG Y rev 1 documentation |
|  sourceType | refer to SEG Y rev 1 documentation |
|  subweatheringVelocity | refer to SEG Y rev 1 documentation |
|  surfaceElevationAtSource | refer to SEG Y rev 1 documentation |
|  sweepFrequencyAtEnd | refer to SEG Y rev 1 documentation |
|  sweepFrequencyAtStart | refer to SEG Y rev 1 documentation |
|  sweepLengthInMsec | refer to SEG Y rev 1 documentation |
|  sweepTaperLenghtAtEndMsec | refer to SEG Y rev 1 documentation |
|  sweepTaperLengthAtStartMsec | refer to SEG Y rev 1 documentation |
|  sweepType | refer to SEG Y rev 1 documentation |
|  taperType | refer to SEG Y rev 1 documentation |
|  timeBasis | refer to SEG Y rev 1 documentation |
|  totalStaticMsec | refer to SEG Y rev 1 documentation |
|  TraceHeaderBuffer | SEG Y TraceHeader storage block |
|  traceIdentificationCode | refer to SEG Y rev 1 documentation |
|  traceNumberWithinEnsemble | refer to SEG Y rev 1 documentation |

| | |
|--|------------------------------------|
|  traceNumberWithinOriginalFieldRecord | refer to SEG Y rev 1 documentation |
|  traceSequenceNumberWithinFile | refer to SEG Y rev 1 documentation |
|  traceSequenceNumberWithinLine | refer to SEG Y rev 1 documentation |
|  traceValueMeasurementUnit | refer to SEG Y rev 1 documentation |
|  traceWeightingFactor | refer to SEG Y rev 1 documentation |
|  transductionConstantExponent | refer to SEG Y rev 1 documentation |
|  transductionConstantMantissa | refer to SEG Y rev 1 documentation |
|  transductionUnits | refer to SEG Y rev 1 documentation |
|  upholeTimeAtGroupMsec | refer to SEG Y rev 1 documentation |
|  upholeTimeAtSourceMsec | refer to SEG Y rev 1 documentation |
|  waterDepthAtGroup | refer to SEG Y rev 1 documentation |
|  waterDepthAtSource | refer to SEG Y rev 1 documentation |
|  weatheringVelocity | refer to SEG Y rev 1 documentation |
|  xCoordinateOfEnsemble | refer to SEG Y rev 1 documentation |
|  yCoordinateOfEnsemble | refer to SEG Y rev 1 documentation |
|  yearDataRecorded | refer to SEG Y rev 1 documentation |

Methods

| | Name | Description |
|---|--------------------------|--|
|  | Equals | Determines whether the specified Object is equal to the current Object . (Inherited from Object .) |
|  | Finalize | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .) |


| | | |
|---|---------------------------------|--|
|  | GetHashCode | Serves as a hash function for a particular type. (Inherited from Object .) |
|  | GetType | Gets the Type of the current instance. (Inherited from Object .) |
|  | Initialize | initialize object |
|  | MemberwiseClone | Creates a shallow copy of the current Object . (Inherited from Object .) |
|  | ToString | Returns a string that represents the current object. (Inherited from Object .) |

SEGYUtilities Class



























SEGYUtilities for use in reading and writing SEG Y files

The **SEGYUtilities** type exposes the following members.

Constructors

| | Name | Description |
|---|-------------------------------|--|
|  | SEGYUtilities | Initializes a new instance of the SEGYUtilities class |

Methods

| | Name | Description |
|--|---------------------------------------|--|
|   | Bytes2Int | convert bytes to long int |
|   | ConvertAsciiToEbcDic | convert an ASCII byte array to an EBCDIC byte array |
|   | ConvertEbcDicToAscii | convert an EBCDIC byte array to an ASCII byte array |
|   | convertPositionToInt | convert a position to a SEG Y trace header integer |
|   | convertToPosition | convert a SEG Y trace header positional value to position |
|   | decimalDegreesToDMS | convert decimal degrees to degrees-minutes-seconds |
|   | degreesToSecondsOfArc | convert decimal degrees to seconds of arc |
|   | dmsToDecimalDegrees | convert degrees-minutes-seconds to decimal degrees |
|  | Equals | Determines whether the specified Object is equal to the current Object . (Inherited from Object .) |
|  | Finalize | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object .) |
|  | GetHashCode | Serves as a hash function for a particular type. (Inherited from Object .) |
|  | GetType | Gets the Type of the current instance. (Inherited from Object .) |
|   | Int2Bytes | convert a long int to bytes |
|  | MemberwiseClone | Creates a shallow copy of the current Object . (Inherited from Object .) |
|   | secondsOfArctoDegrees | convert seconds of arc to decimal degrees |
|  | ToString | Returns a string that represents the current object. (Inherited from Object .) |

Extending Class for Local Variants to SEGY Standard

In the past, some organizations have used fields in the binary file header and/or the binary trace headers to store information not conforming to the published SEGY standard. This class structure can be easily amended by adding a property to the relevant class that gets and sets data from the stored byte array blocks.

For example, to retrieve and set source coordinate X positions from the header, the following property is written in the SEGYTraceHeader class:

```
/// <summary>
/// refer to SEGY rev 1 documentation
/// </summary>
public int sourceCoordinateX
{
    get
    {
        // 72 is the byte location in the header, 4 is the wordlength of an int
        return (int)SEGYUtilities.Bytes2Int(this.iTraceHeaderBuffer, 72, 4, true, isBigEndian);
    }
    set
    {
        SEGYUtilities.Int2Bytes((long)value, true, this.iTraceHeaderBuffer, 72, 4, isBigEndian);
    }
}
```

which uses the SEGYUtilities method, Bytes2Int and Int2Bytes, to retrieve and store this information in the trace header byte array block.

The GSC had stored non-conformant positional information in the trace header in the group coordinate X location in the trace header byte array block. This non-conformant property is retrieved by adding the following to the SEGYTrace class:

```
/// <summary>
/// GSCA implementation of group position
/// </summary>
public double groupPositionXGSCDIG
{
    get
    {
        return SEGYUtilities.convertToPosition(this.iSEGYTraceHeader.groupCoordinateX, 3, -1e6);
    }
    set
    {
    }
}
```

An updated schema can be regenerated using the Microsoft's XML Schema Definition Tool, XSD.exe

Another way of implementing local variants to the SEG-Y standard, without changing the original source code and using the compiled class library, is made possible by implementing a derived class of the top level object, SEG-YFile.

Following the above example,

```
public class derivedSEG-Y : SEG-YFile
{
    /// <summary>
    /// GSCA implementation of group position
    /// </summary>
    public double groupPositionXGSCDIG
    {
        get
        {
            return SEG-YUtilities.convertToPosition(this.currentTrace.TraceHeader.groupCoordinateX, 3, -1e6);
        }
        set
        {
        }
    }
}
```

So in the body of the new code

```
sf2 = new derivedSEG-Y();
sf2.Open(this.openFileDialog1.FileName);
sf2.ReadNextTrace();
double X= sf2.groupPositionXGSCDIG;
```

This approach has the advantage of clearly documenting the institution's variants to the published SEG-Y standard.

References

Barry, K., Cavers, D., and Kneale, C., 1975, Recommended standards for digital tape formats: *Geophysics*, v. 40, p. 344-352.

Norris, M., and Faichney, A., 2002, SEG Y rev 1 Data Exchange format: Technical Standards Committee SEG (Society of Exploration Geophysicists).

Appendix 1 – XML Schema for SEGYlib

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="SEGYFile" nillable="true" type="SEGYFile" />
  <xs:complexType name="SEGYFile">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="isBigEndian" type="xs:boolean" />
      <xs:element minOccurs="0" maxOccurs="1" name="FileHeader" type="SEGYFileHeader" />
      <xs:element minOccurs="0" maxOccurs="1" name="Traces" type="ArrayOfSEGYTrace" />
      <xs:element minOccurs="1" maxOccurs="1" name="NumberOfTracesInBuffer" type="xs:int" />
      <xs:element minOccurs="0" maxOccurs="1" name="currentTrace" type="SEGYTrace" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SEGYFileHeader">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="positionOfStartOfDataTraces" type="xs:long" />
      <xs:element minOccurs="1" maxOccurs="1" name="isSEGYFileHeaderAscii" type="xs:boolean" />
      <xs:element minOccurs="0" maxOccurs="1" name="ExtendedTextHeader" type="ArrayOfBase64Binary" />
      <xs:element minOccurs="0" maxOccurs="1" name="BinaryFileHeader" type="xs:base64Binary" />
      <xs:element minOccurs="1" maxOccurs="1" name="dataSampleFormatCode" type="xs:int" />
      <xs:element minOccurs="1" maxOccurs="1" name="jobIdentificationNumberz" type="xs:unsignedInt" />
      <xs:element minOccurs="1" maxOccurs="1" name="lineNumber" type="xs:unsignedInt" />
      <xs:element minOccurs="1" maxOccurs="1" name="reelNumber" type="xs:unsignedInt" />
      <xs:element minOccurs="1" maxOccurs="1" name="numberOfDataTracesPerEnsemble" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="numberOfAuxiliaryTracesPerEnsemble" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="sampleIntervalInMicroseconds" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="sampleIntervalInMicrosecondsInOriginalFieldRecording" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="numberOfSamplesPerDataTrace" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="numberOfSamplesPerDataTraceForOriginalFieldRecording" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="ensembleFold" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="traceSortingCode" type="xs:short" />
      <xs:element minOccurs="1" maxOccurs="1" name="verticalSumCode" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="sweepFrequencyStart" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="sweepFrequencyEnd" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="sweepLength" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="sweepCode" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="traceNumberSweepChannel" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="sweepTraceTaperLengthAtStart" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="sweepTraceTaperLengthAtEnd" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="taperType" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="correlatedDataTraces" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="binaryGainRecovered" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="amplitudeRecoveryMethod" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="measurementSystem" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="impulseSignalPolarity" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="vibratoryPolarityCode" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="seggyFormatRevisionNumber" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="fixedLengthTraceFlag" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="numberOfExtendedTextualFileHeaderRecordsFollowing" type="xs:unsignedShort" />
      <xs:element minOccurs="1" maxOccurs="1" name="lengthOfFileHeader" type="xs:int" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfBase64Binary">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="base64Binary" nillable="true" type="xs:base64Binary" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfSEGYTrace">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="SEGYTrace" nillable="true" type="SEGYTrace" />
    </xs:sequence>
  </xs:complexType>
```

```

<xs:complexType name="SEGYTrace">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="TraceHeader" type="SEGYTraceHeader" />
    <xs:element minOccurs="0" maxOccurs="1" name="TraceData" type="SEGYTraceData" />
    <xs:element minOccurs="0" maxOccurs="1" name="Data" type="ArrayOfDouble" />
    <xs:element minOccurs="1" maxOccurs="1" name="timeTracedRecorded" type="xs:dateTime" />
    <xs:element minOccurs="1" maxOccurs="1" name="sourcePositionX" type="xs:double" />
    <xs:element minOccurs="1" maxOccurs="1" name="sourcePositionY" type="xs:double" />
    <xs:element minOccurs="1" maxOccurs="1" name="isLatLon" type="xs:boolean" />
    <xs:element minOccurs="1" maxOccurs="1" name="positionOfTraceInFile" type="xs:long" />
    <xs:element minOccurs="1" maxOccurs="1" name="isBigEndian" type="xs:boolean" />
    <xs:element minOccurs="1" maxOccurs="1" name="totalLengthOfTraceData" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="groupPositionXGSCDIG" type="xs:double" />
    <xs:element minOccurs="1" maxOccurs="1" name="groupPositionYGSCDIG" type="xs:double" />
    <xs:element minOccurs="1" maxOccurs="1" name="codedTime" type="xs:long" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="SEGYTraceHeader">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="traceSequenceNumberWithinLine" type="xs:unsignedInt" />
    <xs:element minOccurs="1" maxOccurs="1" name="traceSequenceNumberWithinFile" type="xs:unsignedInt" />
    <xs:element minOccurs="1" maxOccurs="1" name="originalFieldRecordNumber" type="xs:unsignedInt" />
    <xs:element minOccurs="1" maxOccurs="1" name="traceNumberWithinOriginalFieldRecord" type="xs:unsignedInt" />
    <xs:element minOccurs="1" maxOccurs="1" name="energySourcePointNumber" type="xs:unsignedInt" />
    <xs:element minOccurs="1" maxOccurs="1" name="ensembleNumber" type="xs:unsignedInt" />
    <xs:element minOccurs="1" maxOccurs="1" name="traceNumberWithinEnsemble" type="xs:unsignedInt" />
    <xs:element minOccurs="1" maxOccurs="1" name="traceIdentificationCode" type="xs:short" />
    <xs:element minOccurs="1" maxOccurs="1" name="numberOfVerticallySummedTracesYieldingThisTrace" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="numberOfHorizontallySummedTracesYieldingThisTrace" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="dataUse" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="distanceFromCenterOfSourcePointToCenterOfGroup" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="receiverGroupElevation" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="surfaceElevationAtSource" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="sourceDepthBelowSurface" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="datumElevationAtReceiverGroup" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="datumElevationAtSource" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="waterDepthAtSource" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="waterDepthAtGroup" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="scalarForAllElevationsAndDepths" type="xs:short" />
    <xs:element minOccurs="1" maxOccurs="1" name="scalarToBeAppliedToAllCoordinates" type="xs:short" />
    <xs:element minOccurs="1" maxOccurs="1" name="sourceCoordinateX" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="sourceCoordinateY" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="groupCoordinateX" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="groupCoordinateY" type="xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="coordinateUnits" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="weatheringVelocity" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="subweatheringVelocity" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="upholeTimeAtSourceMsec" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="upholeTimeAtGroupMsec" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="sourceStaticCorrectionMsec" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="groupStaticCorrectionMsec" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="totalStaticMsec" type="xs:short" />
    <xs:element minOccurs="1" maxOccurs="1" name="lagTimeAMsec" type="xs:short" />
    <xs:element minOccurs="1" maxOccurs="1" name="lagTimeBMsec" type="xs:short" />
    <xs:element minOccurs="1" maxOccurs="1" name="delayRecordingTimeMsec" type="xs:short" />
    <xs:element minOccurs="1" maxOccurs="1" name="muteTimeStartTimeMsec" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="muteTimeEndTimeMsec" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="numberOfSamplesInTrace" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="sampleIntervalUsec" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="gainTypeOfFieldInstruments" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="instrumentGainConstantDB" type="xs:short" />
    <xs:element minOccurs="1" maxOccurs="1" name="instrumentEarlyOrInitialGainDB" type="xs:short" />
    <xs:element minOccurs="1" maxOccurs="1" name="correlated" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="sweepFrequencyAtStart" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="sweepFrequencyAtEnd" type="xs:unsignedShort" />
    <xs:element minOccurs="1" maxOccurs="1" name="sweepLengthInMsec" type="xs:unsignedShort" />
  </xs:sequence>
</xs:complexType>

```

```

<xs:element minOccurs="1" maxOccurs="1" name="sweepType" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="sweepTaperLengthAtStartMsec" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="sweepTaperLengthAtEndMsec" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="taperType" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="aliasFrequencyHz" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="aliasFilterSlopeDBOctave" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="notchFrequencyHz" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="notchFilterSlopeDBOctave" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="lowCutFrequencyHz" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="highCutFrequencyHz" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="lowCutSlopeDBOctave" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="highCutSlopeDBOctave" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="yearDataRecorded" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="dayOfYear" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="hourOfDay" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="minuteOfHour" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="secondOfMinute" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="timeBasis" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="traceWeightingFactor" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="geophoneGroupNumberOfRollSwitchPositionOne" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="geophoneGroupNumberOfTraceNumberOneWithinOriginalFieldRecord"
type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="geophoneGroupNumberOfLastTraceWithinOriginalFieldRecord"
type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="gapSize" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="overTravel" type="xs:unsignedShort" />
<xs:element minOccurs="1" maxOccurs="1" name="xCoordinateOfEnsemble" type="xs:int" />
<xs:element minOccurs="1" maxOccurs="1" name="yCoordinateOfEnsemble" type="xs:int" />
<xs:element minOccurs="1" maxOccurs="1" name="inLineNumber3D" type="xs:unsignedInt" />
<xs:element minOccurs="1" maxOccurs="1" name="crossLineNumber3D" type="xs:unsignedInt" />
<xs:element minOccurs="1" maxOccurs="1" name="shotpointNumber" type="xs:unsignedInt" />
<xs:element minOccurs="1" maxOccurs="1" name="scalarAppliedToShotPointNumber" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="traceValueMeasurementUnit" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="transductionConstantMantissa" type="xs:int" />
<xs:element minOccurs="1" maxOccurs="1" name="transductionConstantExponent" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="transductionUnits" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="deviceTraceIdentifier" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="scalarUsedToScaleTraceHeaderMSecTimes" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="sourceType" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="sourceEnergyDirectionMantissa" type="xs:int" />
<xs:element minOccurs="1" maxOccurs="1" name="sourceEnergyDirectionExponent" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="sourceMeasurementMantissa" type="xs:int" />
<xs:element minOccurs="1" maxOccurs="1" name="sourceMeasurementExponent" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="sourceMeasurementUnit" type="xs:short" />
<xs:element minOccurs="1" maxOccurs="1" name="bigEndian" type="xs:boolean" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="SEGYZTraceData">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="TraceDataBuffer" type="xs:base64Binary" />
<xs:element minOccurs="0" maxOccurs="1" name="Data" type="ArrayOfDouble" />
<xs:element minOccurs="0" maxOccurs="1" name="DataCopy" type="ArrayOfDouble" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfDouble">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="double" type="xs:double" />
</xs:sequence>
</xs:complexType>
<xs:element name="SEGYZFileHeader" nillable="true" type="SEGYZFileHeader" />
<xs:element name="SEGYZTrace" nillable="true" type="SEGYZTrace" />
<xs:element name="SEGYZTraceData" nillable="true" type="SEGYZTraceData" />
<xs:element name="SEGYZTraceHeader" nillable="true" type="SEGYZTraceHeader" />
<xs:element name="SEGYZUtilities" nillable="true" type="SEGYZUtilities" />
<xs:complexType name="SEGYZUtilities" />
</xs:schema>

```