

GXTF

GEOMLIB (β)

**MARINE ENGINEERING GEOPHYSICAL
DATA PROCESSING TOOLBOX**

Ivan V. Dmitriev
03.08.2020

Contents

1	Xtf files read and write.....	3
1.1	Message Header	5
1.2	000=Sonar Data Message.....	5
1.3	Examples	6
2	Xtf additional functions.....	9
2.1	Delete Messages from Xtf-structure.....	9
2.2	Append files	9
2.3	Delete Slant-range data for Sonar Data Message	9
2.4	Create Poly-line from files with Sonar Data Message	10
2.5	Interpolate fields from Head1 to fields for Head2 with Message Type 000	11
2.6	Create DTEN fields (coordinates transformation) 30/07/2020	11
2.7	Re-calculate Xtf-fields using DTEN fields 30/07/2020	12
2.8	Create Dataset from Xtf-files with Message Type 000 03/08/2020.....	13
2.9	Insert Dataset with Message Type 000 to Xtf-files 03/08/2020.....	13
	Citation.....	15

Figures list

Figure 1	Xtf records structure	4
Figure 2	gXtf using (example)	8

1 Xtf files read and write

MatLab functions set for reading and writing Xtf-files (*Table 1*). Based on the paper “Triton Imaging, Inc. eXtended Triton Format (XTF) Rev. 35”.

Table 1 gXtf functions

Function name	Function description
gXtfHeaderRead	Read XtfHead structure from *.xtf file.
gXtfHeaderWrite	Write XtfHead structure to new *.xtf file.
gXtfFileSetMessOff	Mark messages typed in Head to Delete from Xtf structure.
gXtfFilesAppend	Append Xtf files in order, using Header data.
gXtf000Read	Read [Head,Data] from XtfHead.fName (*.xtf) file for Message Type 000
gXtf000Write	Write [XtfHead,Head,Data] to *.xtf file for Message Type 000.
gXtf000Dir2PL	Read coordinates form SSS files (Message Type 000) to PL-structure.
gXtf000DeleteSlant	Delete slant-range data in Xtf-structure for Message Type 000
gXtf000Field2Field	Interpolate fields from Head1 to fields for Head2 with Message Type 000
gXtfDTEN	Create Nav's fields GpsDay,GpsTime,GpsE,GpsN,GpsH from Xtf-fields
gXtfDTENinv	Create Xtf-fields from Nav's fields GpsDay,GpsTime,GpsE,GpsN,GpsH
gXtf000DatasetImport	Create Dataset from Xtf-files with Message Type 000
gXtf000DatasetExport	Insert Dataset with Message Type 000 to Xtf-files

Xtf-file includes the File Header and Data Blocks (*Figure 1*). File Header consists of two structures XtfFileHeader and ChanInfo. The number of channels in ChanInfo structure can be different and assigned in XtfFileHeader structure.

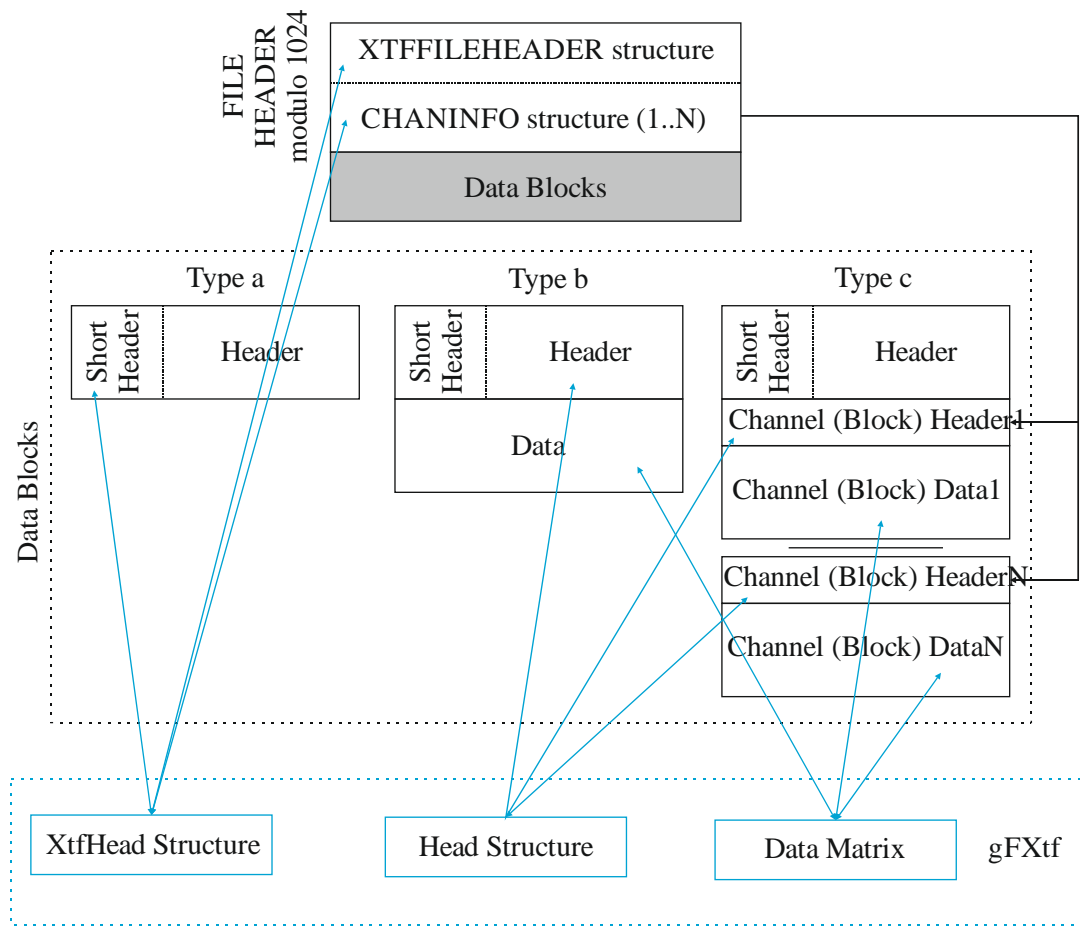


Figure 1 Xtf records structure

There are three types of Data Blocks:

- a) Header only (structured data with fields);
- b) Header with follow Data (where Data is the element of big numeric matrix, the same MBES ping);
- c) Header with follow chain Channel's Header and Channel's Data (where Data is the element of big numeric matrix, the same sonar ping, as part of channel data).

XtfHead and Head structures

XtfHead include XTFFILEHEADER, CHANINFO and Short Header structures from xtf-file.

The Short Header defined in the gFXtf, as a piece of Head. The Short Header used for navigation in the file; Short Header not load in gFXtf Head Structure (to avoid data duplication). It does include the first six fields, general for most Headers: 1) XtfHead.RFace; 2) XtfHead.RHeaderType; 3) XtfHead.RSubChannelNumber; 4) XtfHead.RNumChansToFollow; 5) XtfHead.RUnused; 6) XtfHead.RNumBytesThisRecord.

XtfHead include the next addition fields:

XtfHead.Descript – description for some types of structure fields;

XtfHead.fName – name of Xtf-file for that structure.

XtfHead.RSeek – File pointer in Xtf-file to each Header Block part, the next after Short Header.

XtfHead.ROnFlag – Flag WriteOn for each Head or [Head,Data] (depending on context).

1.1 Message Header

function XtfHead=gXtfHeaderRead (fName,flStat)

Read XtfHead structure (include XTFFILEHEADER, CHANINFO and Short Header) from *.xtf file.

Parameters:

XtfHead – XtfHead structure;

fName – the target file name;

flStat – flag for statistics display (1 or 0).

Example: `XtfHead=gXtfHeaderRead ('c:\temp\1.xtf',1);`

function gXtfHeaderWrite (XtfHead,fNameNew)

Write XtfHead structure (include XTFFILEHEADER, CHANINFO and Short Header) to new *.xtf file.

Parameters:

XtfHead – structure with parent xtf-file description;

fNameNew – string, the target file for writing.

gFXtfWriteHeader used file XtfHead.fName as data source. The file XtfHead.fName must be presented; names XtfHead.fName and fNameNew must be different.

You can use XtfHead.ROnFlag to except any record from new ftx-file (see gXtfHeaderMessOff).
Warning: for new file CHANINFO Structure not controlled; if necessary, remove Chanel Records in handle.

Example: `gXtfHeaderWrite (XtfHead,'c:\temp\1new.xtf');`

1.2 000=Sonar Data Message

function [Head,Data]=gXtf000Read(XtfHead,SubCh)

Read [Head,Data] from XtfHead.fName (*.xtf) file for Message Type 000 (Sonar Data Message).

Parameters:

XsfHead – Xsf Header structure;

SubCh – sub channel number;

Head – Header structure;

Data – Data Body for sonar channels.

Head include the next addition fields: Head.HSubChannelNumber, Head.HMessageNum.

Example: `[Head,Data]=gXtf000Read(XtfHead,0);`

Warning! There is any lawlessness for SubChannelNumber corresponding between CHANINFO structure and Channel's Header. For two channels xtf-file

C-Max >>> XtfHead.CSubChannelNumber=[0 1] and XtfHead.RSubChannelNumber(:)=0;

EdgeTech4200 >>> XtfHead.CSubChannelNumber=[0 0] and XtfHead.RSubChannelNumber(:)=0;

Qinsy >>> XtfHead.CSubChannelNumber=[0 1] and XtfHead.RSubChannelNumber(:)=1.

function gXtf000Write(XtfHead,Head,Data,fNameNew,flTraceLenChanged)

Write [XtfHead,Head,Data] to *.xtf file for Message Type 000 (Sonar Data Message).

Parameters:

XsfHead – Xsf Header structure;

Head – Header structure;

Data – Data Body for sonar channels;

fNameNew – string, the target file for writing;

flTraceLenChanged – key for changes:

flTraceLenChanged=0, if Data Trace Length (Head.CNumSamples) was not changed;

flTraceLenChanged=1, if Data Trace Length (Head.CNumSamples) was changed (see gXtf000DeleteSlant) and RNumBytesThisRecord field need to correct;

flTraceLenChanged=2, if file XtfHead.fName is absent;

%gXtf000Write used file XtfHead.fName as data source for XtfHead.RHeaderType~=0. The file XtfHead.fName must be presented (flTraceLenChanged=2 is exception); names XtfHead.fName and fNameNew must be different.

The XtfHead is description for XtfHead.fName file. Only XtfHead.ROnFlag can be changed to this structure. XtfHead.ROnFlag can be used to except any record for new xtf-file.

If you change Data-Trace Length (Head.CNumSamples), the RNumBytesThisRecord field must be correct. Use flTraceLenChanged==1 for correction.

Example: `gXtf000Write(XtfHead,Head,Data,'c:\temp\1new.xtf',0);`

1.3 Examples

Example 1

%Read XtfHead for Qinsy and show statistic

```
>> XtfHead=gXtfHeaderRead ('c:\temp\1.xtf',1);
```

```
Rec: 0=XTF_HEADER_SONAR (Sidescan data), Num: 11079 [ SubCh: 1, Num: 11079; ChFollow: 2, Num: 11079 ]
```

```
Rec: 66=XTF_HEADER_QINSY_R2SONIC_FTS QINSy R2Sonics Foot Print Time Series (snippets), Num: 11079 [ SubCh: 2, Num: 11079; ChFollow: 0, Num: 11079 ]
```

```
Rec: 100=XTF_HEADER_POSITION Raw position packet - Reserved for use by Reson, Inc. RESON ONLY, Num: 3049 [ SubCh: 1, Num: 3049; ChFollow: 0, Num: 3049 ]
```

```

% Save xtf-file with XtfHead
>> gXtfHeaderWrite (XtfHead,'c:\temp\1new.xtf');
% Set XtfHead.ROnFlag at 0 for Header Type 100
XtfHead.ROnFlag(XtfHead.RHeaderType ==100)=0;
% Save xtf-file with XtfHead (without Header Type 100)
>> gXtfHeaderWrite (XtfHead,'c:\temp\2new.xtf');
%Read XtfHead from new file and show statistic
>> XtfHeadNew=gXtfHeaderRead ('c:\temp\2new.xtf',1);
Rec: 0=XTF_HEADER_SONAR (Sidescan data), Num: 11079 [ SubCh: 1, Num: 11079; ChFollow: 2,
Num: 11079 ]
Rec: 66=XTF_HEADER_QINSY_R2SONIC_FTS QINSY R2Sonics Foot Print Time Series (snippets),
Num: 11079 [ SubCh: 2, Num: 11079; ChFollow: 0, Num: 11079 ]

```

Example 2

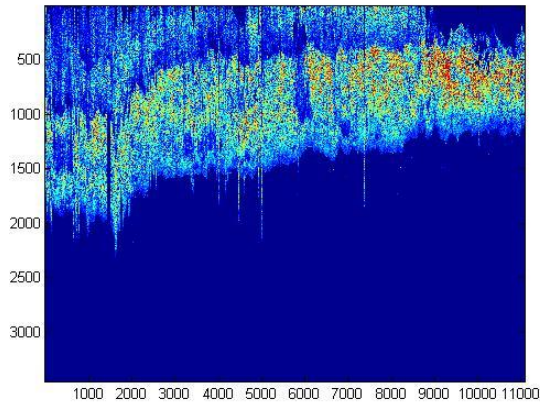
```

%Read XtfHead without statistic
>> XtfHead=gXtfHeaderRead ('c:\temp\Qinsy0001.xtf',0);
%Read sonar channel (Header Type 000)
>> [Head,Data]=gXtf000Read(XtfHead,1);
%Draw sonar channel 1 data (Figure 2, a)
>> imagesc(Data(:,1));
%Set part of Data to 50000 for testing
>> Data(:,1:4000,1)=5e4;
% Write XtfRecords with Header Type 000 to a new file
>> gXtf000Write(XtfHead,Head,Data,'c:\temp\Qinsy0001.xtf',0);
% Read new XtfHead without statistic
>> XtfHeadNew=gXtfHeaderRead ('c:\temp\Qinsy0001.xtf',0);
%Read new sonar channel (Record Type 80)
>> [HeadNew,DataNew]=gXtf000Read(XtfHeadNew,1);
%Draw new sonar channel 1 data (Figure 2, b)
>> imagesc(Data(:,1));
%Draw sonar track plot (Figure 2, c)
>> plot(Head.HShipYcoordinate./nnn,Head.HShipXcoordinate./nnn);
%Draw sonar SlantRange (Figure 2, d)
>> plot([Head.CSlantRange(:,1) Head.CSlantRange(:,2)+3]);

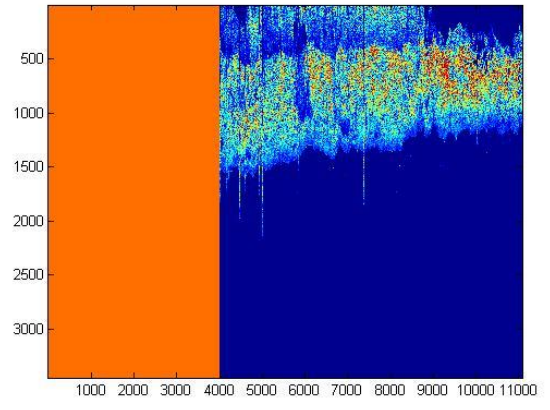
```

(a)

(b)



(c)



(d)

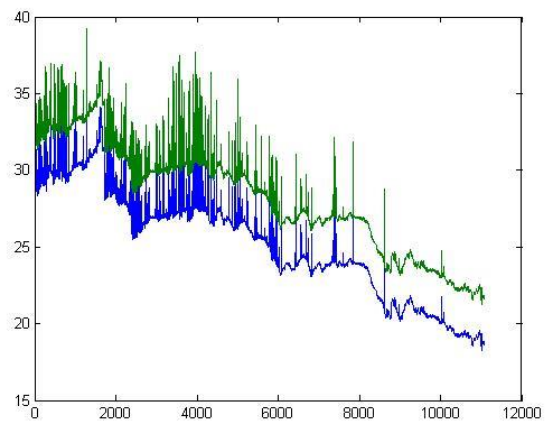
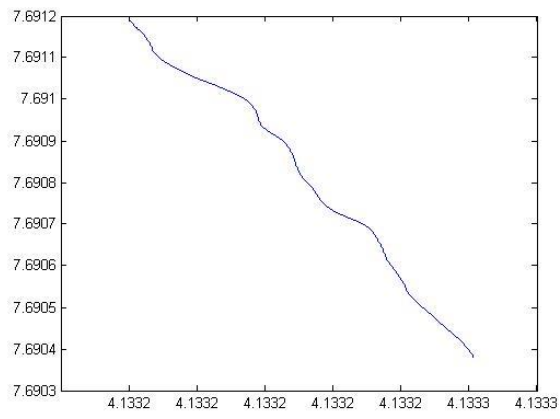


Figure 2 gXtf using (example)

2 Xtf additional functions

2.1 Delete Messages from Xtf-structure

function XtfHead=gXtfFileSetMessOff(XtfHead,Head,mask)

Mark messages typed in Head to Delete from Xtf structure; need write file with gXtfWriteHeader for Deleting.

Parameters:

[XtfHead,Head] – Xtf structures;

mask – mask for messages marked off/on (logical or trace_numbers);

XtfHead – Xtf structure with messages switched off.

Function Example:

```
>> XtfHead=gXtfHeaderRead ('e:\ND49_V17B1212_SSSH.xtf',1);
>> [Head,Data]=gXtf000Read(XtfHead,0);
>> mask=logical(ones(1,size(Data,2)));mask(1:4000)=0;
>> XtfHead=gXtfFileSetMessOff(XtfHead,Head,mask);
>> gXtfHeaderWrite (XtfHead,'e:\ND49_V17B1212_SSSH.z.xtf');
>> XtfHead= gXtfHeaderMessOff (XtfHead,Head,1:6000);
>> gXtfHeaderWrite (XtfHead,'e:\ND49_V17B1212_SSSH.z.xtf');
```

2.2 Append files

function gXtfFilesAppend(fNameNew,varargin)

Append Xtf files in order, using Header data.

Parameters:

fNameNew – file name for appending result;

varargin{1..N} – file names will be appending.

The XTFFILEHEADER Structure and CHANINFO Structure get from file varargin{1}. It must be equal for all appending files. ShortHeader structures and Data Blocks appended from all files.

Example:

```
gXtfFilesAppend ('c:\temp\App.xtf','c:\temp\1.xtf','c:\temp\2.xtf','c:\temp\3.xtf');
```

2.3 Delete Slant-range data for Sonar Data Message

function [Head,Data]=gXtf000DeleteSlant(Head,Data,mask)

Delete slant-range data in Xtf-structure for Message Type 000 (Sonar Data Message).

Parameters:

[Head,Data] – Xtf structure // Data must contained traces with equal length;
 mask – mask for slant-range data delete (logical or trace_numbers);
 [Head,Data] – Xtf structure with deleted slant-range data.

Function Example:

```
>> XtfHead=gXtfHeaderRead ('e:\ND49_V17B1212_SSSH.xtf',1)
>> [Head,Data]=gXtf000Read(XtfHead,0);
>> figure(1);imagesc(Data(:,1));figure(2);imagesc(Data(:,2));
>> [Head,Data]=gXtf000DeleteSlant(Head,Data,1:2000);
>> figure(3);imagesc(Data(:,1));figure(4);imagesc(Data(:,2));
>> gXtf000Write(XtfHead,Head,Data,'e:\ND49_V17B1212_SSSHzz.xtf',1);
```

2.4 Create Poly-line from files with Sonar Data Message

function PL=gXtf000Dir2PL(fName,KeyLineDraw,FieldKP,NavS,NavP,NavPTargCode)

Read coordinates form Directory with SSS files (Message Type 000) to PL-structure.

Parameters:

fName – reading file name or folder name with files (last name's symbol must be '\');

keyLineDraw- string key for line drawing: '-r','xb', etc;

FieldKP – field from Head structure (gFSgyRead function) copied to PL(n).pKP; if empty, than
 PL(n).pKP=1:length(PL(n).pX).

NavS – (see gNavCoord2Coord) navigation datum for Sensor, fields: EllipParam, ProjParam,
 ProjForvFunc, ProjRevFunc, EllipTransParam, EllipForvTransFunc, EllipRevTransFunc,
 TargCode.

if ~isfield(NavS.EllipTransParam), then transformation Sensor's_Ellipsoid-to-Project's ellipsoid not
 calculate (fields EllipTransParam, EllipForvTransFunc, EllipRevTransFunc not used).

NavP – (see gNavCoord2Coord) navigation datum for Project, fields: EllipParam, ProjParam,
 ProjForvFunc, ProjRevFunc, TargCode.

NavPTargCodes= forced output_datum_code (see gNavCoord2Coord); there are: 1) sensor XY;
 2) sensor geographic; 3) sensor geocentric; 4)project geocentric; 5)project geographic; 6)project
 XY.

if isempty(NavPTargCodes), than create NavPTargCodes=NavP.TargCode

if isempty(NavS)&&isempty(NavP)&&isempty(NavPTargCode), than there is no any coordinate
 transformation;

PL – output structure: PL(n).PLName; PL(n).Type; PL(n).KeyLineDraw; PL(n).GpsE; PL(n).GpsN;
 PL(n).GpsH; PL(n).GpsKP (to GpsKP write ping number in file).

Example:

```
>> NavS=struct('TargCode',2);
```

```
>> NavP=struct('EllipParam',[6378137 0.081819190842], 'ProjParam',[0 142 0.9996 500000 0],
'ProjForvFunc','gNavGeog2ProjUtm', 'ProjRevFunc','gNavProjUtm2Geog', 'TargCode',6);
>> PLXtf=gXtf000Dir2PL('c:\temp\SSS\3\','-b',[,NavS,NavP,6,[]]);
>> gMapPLDraw(100,PLXtf,'PLName');axis equal;
```

2.5 Interpolate fields from Head1 to fields for Head2 with Message Type 000

function Head2=gXtf000Field2Field(Head1,fi1,Head2,fi2)

Interpolate fi1-field from Head1 to fi2-field for Head2 using time-fields (HYear,HMonth,HDay,HHour,HMinute,HSecond,HHSeconds).

Parameters:

Head1- head-structure used as data source;

fi1- field name (chars vectors or cells with chars vectors) of Head1, used as data source;

Head2- head-structure used as data receiver;

fi1- field name (chars vectors or cells with chars vectors) of Head2, used as data receiver;

The data linear interpolated/extrapolated from Head1.(fi1) to Head2.(fi2) using time-fields (HYear,HMonth,HDay,HHour,HMinute,HSecond,HHSeconds).

Example:

```
>> XtfHead1=gXtfHeaderRead('g:\zzzzz\0300_20_21(00)H_sn.xtf',1);
>> [Head1,Data1]=gXtf000Read(XtfHead1,0);
>> XtfHead2=gXtfHeaderRead('g:\zzzzz\0300_20_21(00)L.xtf',1);
>> [Head2,Data2]=gXtf000Read(XtfHead2,0);
>> Head2=gXtf000Field2Field(Head1,{'HSensorYcoordinate','HSensorXcoordinate'},Head2,
{'HSensorYcoordinate','HSensorXcoordinate'});
>> gXtf000Write(XtfHead2,Head2,Data2,'g:\zzzzz\0300_20_21(00)L_sn.xtf',0);
```

2.6 Create DTEN fields (coordinates transformation)

30/07/2020

function Head=gXtfDTEN(XtfHead,Head,FieldKP,FieldX,FieldY,NavS,NavP, HNavUnitsForce, varargin)

Create Nav's fields GpsDay,GpsTime,GpsE,GpsN,GpsH from Xtf-files fields (different for Messages Types)

Parameters:

[XtfHead,Head] – Xtf structure;

FieldKP – the name of Kp-field; there are HMessageNum, HPingNumber;

FieldX,FieldY – the names of X and Y fields; there are Head.HShipYcoordinate, Head.HShipXcoordinate, Head.HSensorYcoordinate, Head.HSensorXcoordinate;

NavS – Sensor's Nav-structure (for segy-file);

NavP – Project's Nav-structure;

HNavUnitsForce- forced HNavUnits for XtfHead (if empty, than no forced changes);

varargin=H – height in Sensor's Nav-structure datum;

Head – output Header with fields: GpsDay,GpsTime,GpsE,GpsN,GpsH.

Used functions: gNavTime2Time,gNavDayCheck,gNavCoord2Coord.

Example:

```
>> Head=gXtfDTEN(XtfHead,Head,'HPingNumber','HShipYcoordinate',...  
    'HShipXcoordinate',NavS,NavP,[],H);
```

2.7 Re-calculate Xtf-fields using DTEN fields

30/07/2020

**function[XtfHead,Head,varargout]=gXtfDTENinv(XtfHead,Head,FieldKP,FieldX,FieldY,NavS,
 NavP,HNavUnitsForce)**

Use Nav's fields [GpsDay,GpsTime,GpsE,GpsN,GpsH,FieldKP] from Head's-Xtf-files-fields to calculate [(FieldX),(FieldY),HYear,HJulianDay,HHour,HMinute,HSecond,HHSeconds,(FieldKP)]

Parameters:

[XtfHead,Head] – Xtf structure;

FieldKP – the name of Kp-field; there are HMessageNum, HPingNumber

FieldX,FieldY – the names of X and Y fields; there are

Head.HShipYcoordinate,Head.HShipXcoordinate,Head.HSensorYcoordinate,Head.HSensorXcoordinate;

NavS – Sensor's Nav-structure (for xtf-file);

NavP – Project's Nav-structure;

HNavUnitsForce – forced HNavUnits for XtfHead (if empty, than no forced changes);

XtfHead – output XtfHead with updated XtfHead.HNavUnits;

Head – output Header with updated fields

[(FieldX),(FieldY),HYear,HJulianDay,HHour,HMinute,HSecond,HHSeconds,(FieldKP)];

varargout=H – output height in Sensor's Nav-structure datum;

Used functions: gNavTime2Time,gNavDayCheck,gNavCoord2Coord.

Example:

```
>> [XtfHead,Head]=gXtfDTENinv(XtfHead,Head,'HPingNumber','HShipYcoordinate',...  
    'HShipXcoordinate',NavS,NavP,[],H);
```

2.8 Create Dataset from Xtf-files with Message Type 000

03/08/2020

function [XtfHead,Head,Data]=gXtf000DatasetImport(fName,tmpName,XtfHead,SubCh,Head,Data,FieldKP,FieldX,FieldY,NavS,NavP,HNavUnitsForce)

Add xtf-files for Message Type 000 (Sonar Data Message) from file-list or folder to Dataset.

Parameters:

fName – list with file's names or folder name with xtf (Message Type 000 - Sonar Data Message) will be loaded;

tmpName – folder name for temporary files saving; if isempty, than Data will be empty;

XtfHead – input XtfHead(1..n) structure (can be empty);

SubCh – sub channel number;

Head – input Head(1..n) structure (can be empty);

Data – input cells with Data-matrix or temporary file names;

FieldKP – the name of Kp-field; there are HMessageNum, HPingNumber

FieldX,FieldY – the names of X and Y fields; there are Head.HShipYcoordinate, Head.HShipXcoordinate, Head.HSensorYcoordinate, Head.HSensorXcoordinate;

NavS – sensor's vanigation structure (see gNavCoord2Coord) for coordinates transformation;

NavP – sensor's vanigation structure (see gNavCoord2Coord) for coordinates transformation;

HNavUnitsForce – forced HNavUnits for XtfHead (if empty, than no forced changes);

[XtfHead,Head,Data] – output variables with added data from files;

Additional fields:

XtfHead(n).fNameTmp – name of temporary file with Data matrix;

GpsDay,GpsTime,GpsE,GpsN,GpsH – fields created by function gXtfDTEN.

Example:

```
>> NavS=struct('TargCode',2);NavP=struct('EllipParam',[6378137 0.081819190842],'ProjParam',[0 142  
0.9996 500000 0],'ProjForvFunc','gNavGeog2ProjUtm','ProjRevFunc','gNavProjUtm2Geog',  
'TargCode',6);
```

```
>> [XtfHead,Head,Data]=gXtf000DatasetImport('c:\xtfin\','c:\xtfin\tmp\',XtfHead,0,Head,Data,  
'HPingNumber','HShipXcoordinate','HShipYcoordinate',NavS,NavP,0);
```

```
>> [XtfHead,Head,Data]=gXtf000DatasetImport('c:\xtfin\','c:\xtfin\tmp\',[],0,[],[], 'HPingNumber',  
'HSensorXcoordinate','HSensorYcoordinate',[],[],[]);
```

2.9 Insert Dataset with Message Type 000 to Xtf-files

03/08/2020

function gXtf000DatasetExport(DirName,XtfHead,Head,Data,FieldKP,FieldX,FieldY,NavS,NavP,HNavUnitsForce)

Add Xtf-variables from Dataset for Message Type 000 (Sonar Data Message) to files (GpsKP,GpsDay,GpsTime,GpsE,GpsN,GpsH fields are used).

Parameters:

DirName – folder for export;

XtfHead – input XtfHead(1..n) structure (can be empty);

Head – input Head(1..n) structure (can be empty);

Data – input cells with Data-matrix or temporary file names;

FieldKP – the name of Kp-field; there are HMessageNum, HPingNumber

FieldX,FieldY – the names of X and Y fields; there are

Head.HShipYcoordinate,Head.HShipXcoordinate,Head.HSensorYcoordinate,Head.HSensorXcoordinate;

NavS – sensor's navigation structure (see gNavCoord2Coord) for coordinates transformation;

NavP – project's navigation structure (see gNavCoord2Coord) for coordinates transformation;

HNavUnitsForce – forced HNavUnits for XtfHead (if empty, than no forced changes).

Additional fields:

XtfHead(n).fNameTmp – name of temporary file with Data matrix;

GpsKP,GpsDay,GpsTime,GpsE,GpsN,GpsH – fields with time and coordinates used function
gXtfDTENinv.

Example:

```
>> NavS=struct('TargCode',2);NavP=struct('EllipParam',[6378137 0.081819190842],'ProjParam',[0 142  
0.9996 500000 0],'ProjForvFunc','gNavGeog2ProjUtm','ProjRevFunc','gNavProjUtm2Geog',  
'TargCode',6);  
>> [XtfHead,Head,Data]=gXtf000DatasetImport('c:\xtfin\','c:\xtfin\tmp\',XtfHead,0,Head,Data,  
'HPingNumber','HShipYcoordinate','HShipXcoordinate',NavS,NavP,[],0);  
>> gXtf000DatasetExport('c:\xtfout\',XtfHead,Head,Data,'HPingNumber','HShipYcoordinate',  
'HShipXcoordinate',NavS,NavP,[]);
```

Citation

- 1) Triton Imaging, Inc. eXtended Triton Format (XTF) Rev. 35. 02/04/2013