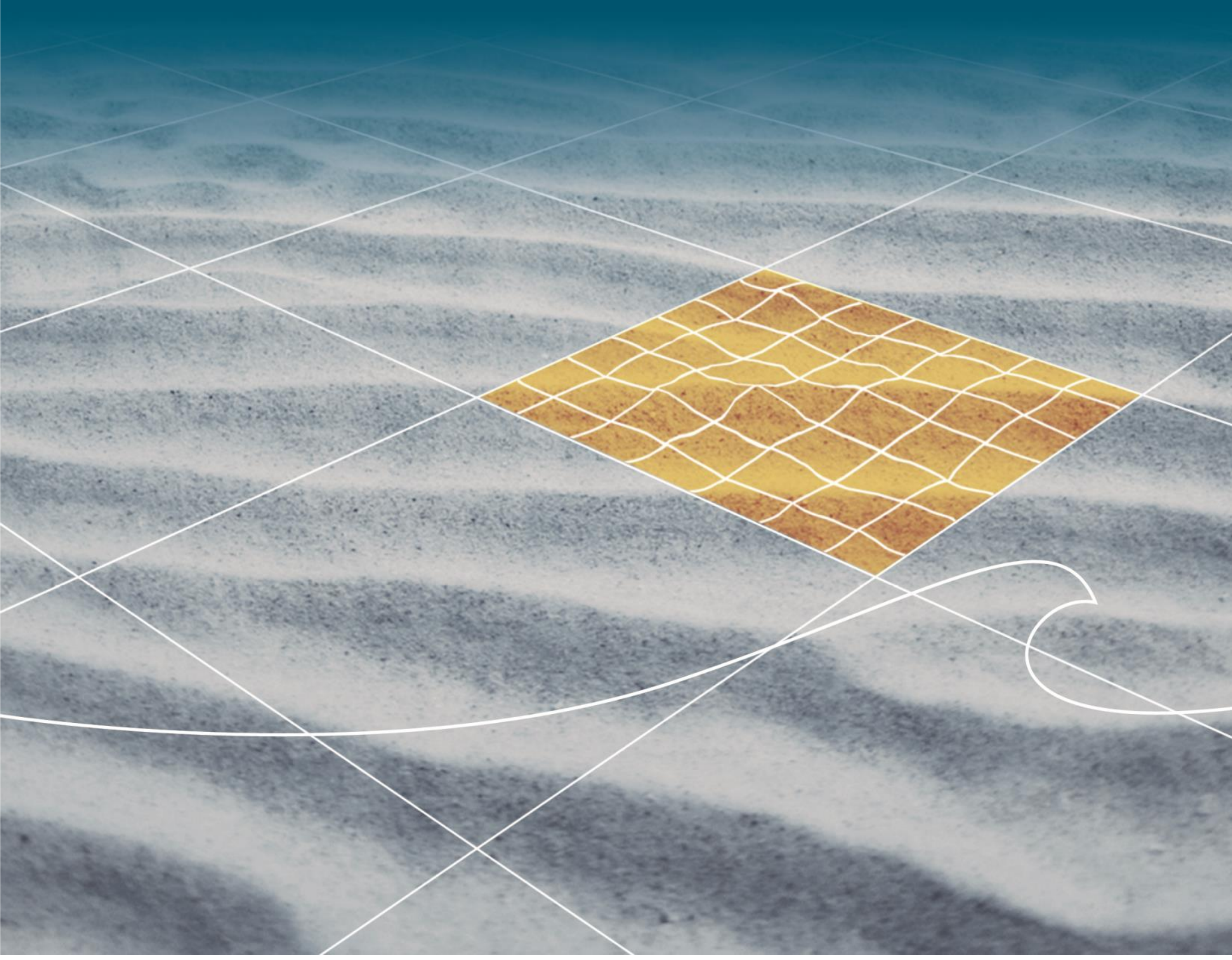
EIVA
MARINE SURVEY SOLUTIONS

# NAVIPAC 4.2

## NAVIPAC LOGGING UTILITY



# Contents

# 1 Introduction

This document describes the NaviPac logging program and the logging formats that are used in the NaviPac product. The design of the program was influenced by the fact that logged data should easily be read by other programs like NaviEdit and spreadsheet programs like Microsoft Excel, among others.

The main task of the NaviPac logging process (**LogData**) is to log data from navigation instruments, and data calculated by the NaviPac Kernel processes.

The program is a client network application that is a part of the NaviPac software package. It can be started from in the online program's **View** menu under the option **Log Data**, or the icon in the Online toolbar as illustrated in the below figure.

Figure 1 Icon for starting data recording

Data can be logged in 3 different formats:

1. A general format that will contain all information, including events and raw instrument data.
2. A XYZ ASCII record format (hard coded) to be read by the NaviEdit interpreter (or another charting package able to read the format). All positions, depths and attitudes (gyro, roll, pitch, heave) values are logged.
3. A format which can be fully customised by the user. The user can select which information and sequence to log into an ASCII file.

All file formats used are ASCII text.

The user can log in all three logging formats at the same time.

**Note**: Formats 1 and 2 are preceded by headers with geodesy and instrument setup information.

**Note**: Events (annotations: eg start/stop/cycle time will not be logged in the XYZ ASCII record format.

**Note**: Raw data will only be logged in general format.

## 1.1 The process environment

The **LogData** program connects to the NaviPac **RemoteAccess** process and thereby continuously receives data after the user selects Start Logging” from the File menu, the toolbar or when logging is started remote from Helmsman.

The program disconnects from the **Kernel (Remote access)** when the user selects **Stop Logging**. In the figure below, the process environment for the logging program is illustrated.

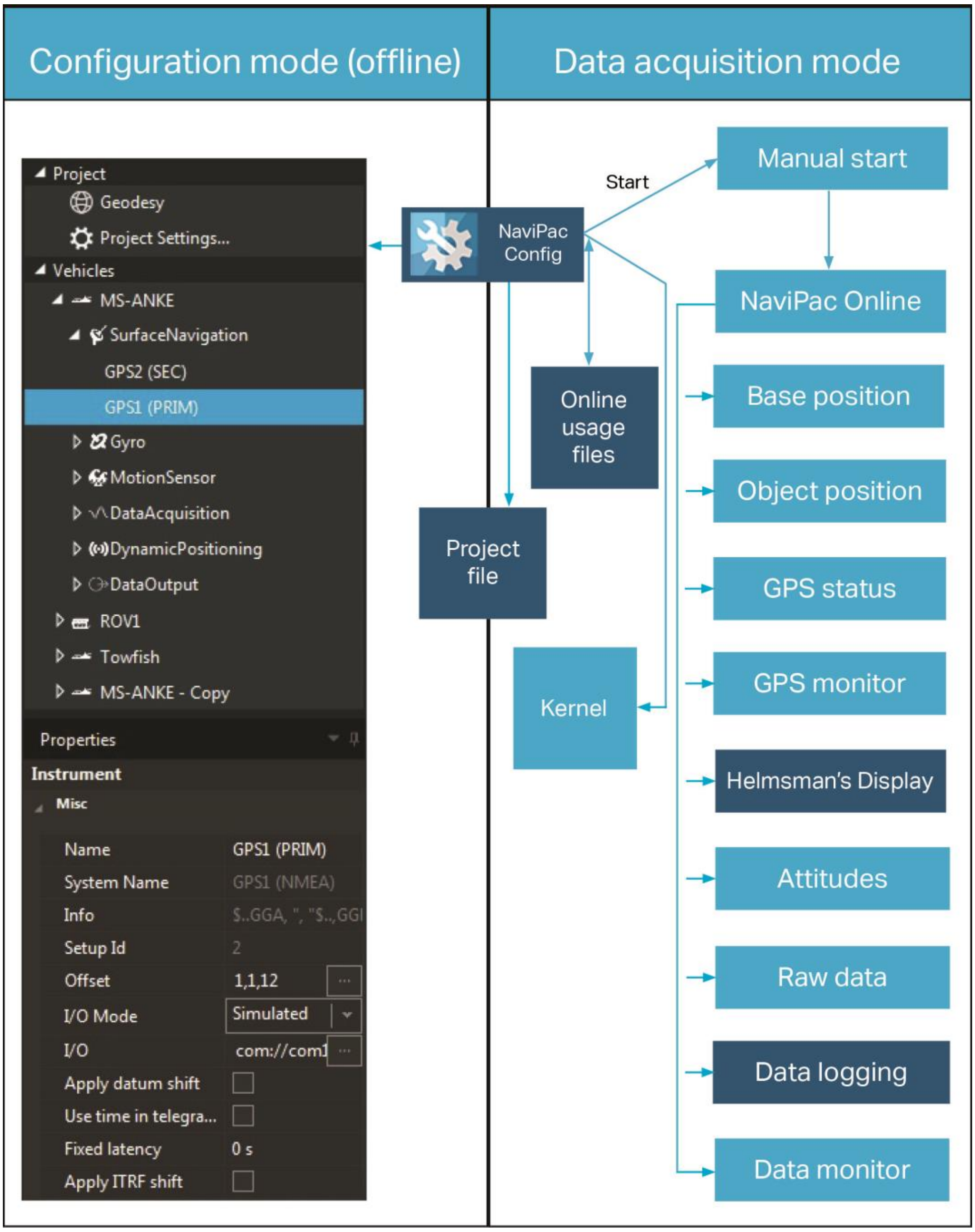



Figure 2 NaviPac process environment

The process uses the TCP/IP protocol in communication with NaviPac (RemoteAccess). This means it can be configured to run on all computers connected in the network, thereby removing workload from the data acquisition computer.

When **LogData** is started, it creates a minimised process window called **NPRec**. It is not included in the above drawing, as it is functionally a part of **LogData**. **NPRec** reads NaviPac data via shared memory, and utilises a much more efficient communication than the TCP/IP.

# 2 The user interface

**LogData** is written in Microsoft Visual C++ 5.0 or later.  The main window currently presents statistical information of the logging process when logging to files is enabled

The main window shows statistic information and the current data that will be logged:

- Current log file names in use (general, custom,  XYZ record)
- Number of position, depth, attitude, gyro, speed records received from Kernel in current log session – ID record counters.
- Snapshot of  actual values received: positions, Gyro, Attitudes, Depths, Speed, Events (not the raw records)

The user can by selecting **File: Set-up Logging** decide what information to see.

**Note**: Every time the users stop/start the logging, new log files are generated.

From the program's GUI (graphical user interface), the user can start and stop the logging, as well as define the custom logging format (or – more specifically – which items to store and which separator to use). In addition, the user can select which formats will be logged.

The video player icons in the toolbar are used to start/stop the logging.

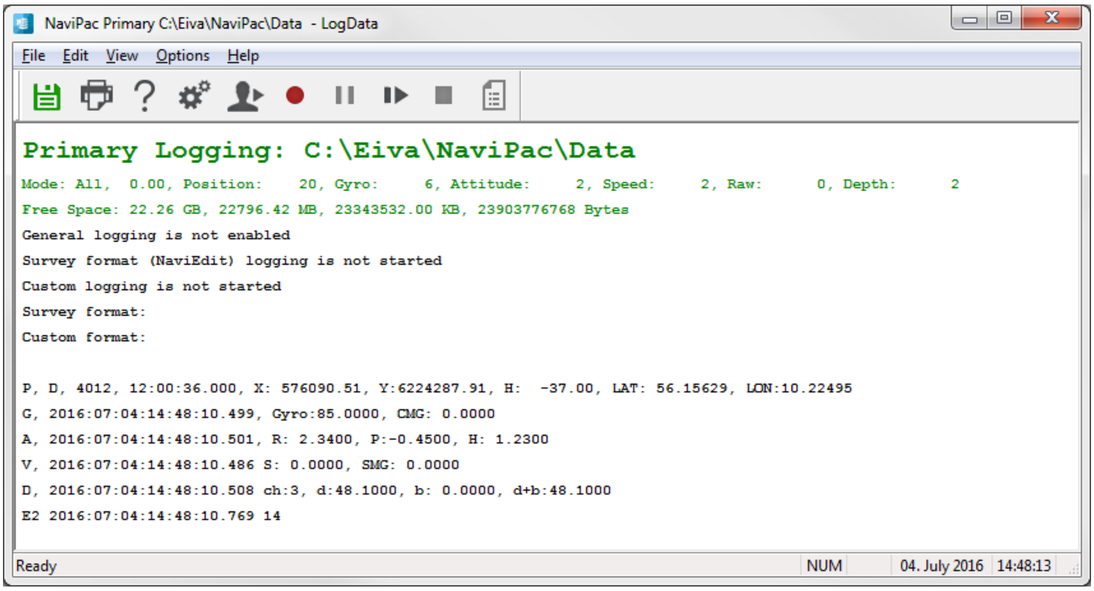


Figure 3 Main Logging window with statistic counters – logging not started

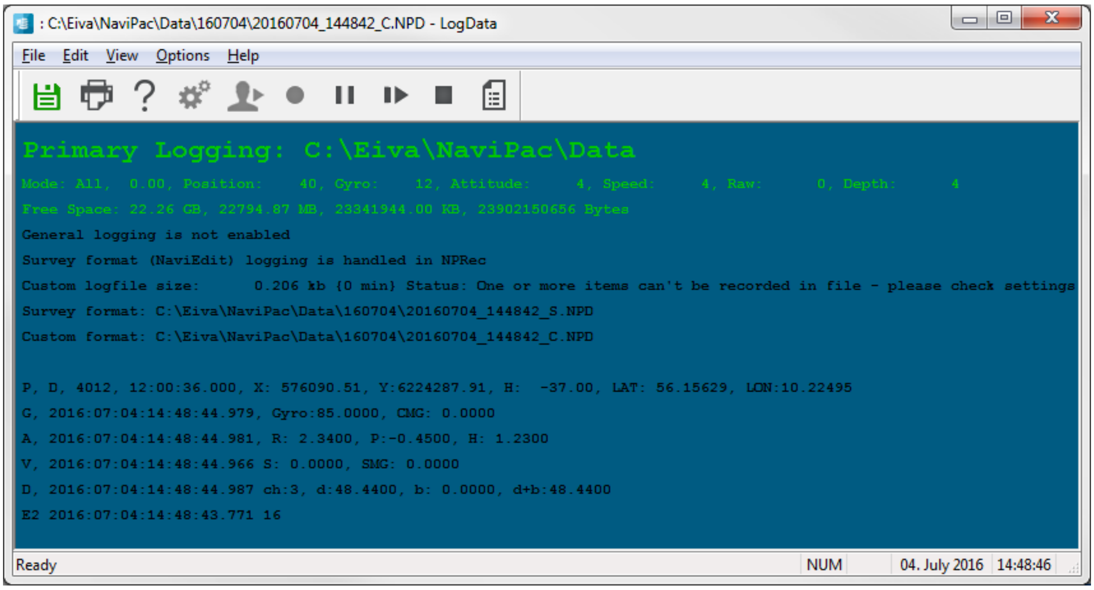


Figure 4 Main Logging window with statistic counters – logging started

In the figure above, a logging session is started with both the custom format and the general format enabled. The user can stop logging by clicking **Stop** icon (the square).

## 2.1 Toolbar

The appearance of the toolbar changes depending on whether or not logging has started, as seen in the following figures:


Figure 5 Toolbar when logging is started

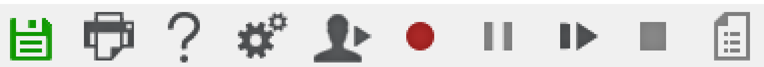


Figure 6 Toolbar when logging is not started

The toolbar holds commonly-used functions from the menus:

- Save Settings
- Print
- Help (version)

- Set up logging
- Custom logging format
- Start logging
- Pause logging
- Stop and start logging
- Stop logging.
- View custom log file

The functionality of these is described in the following sections.

## 2.2 File menu

### 2.2.1 Start Logging

Create new log file(s) (there are 3 log formats possible depending on settings), write the start header information, connect to **Kernel** and begin to receive **log-data**.

**Note**: **Start Logging** can also be done by clicking **Start** from the Helmsman's Display's when a runline is selected. **Start Logging** cannot be started from Helmsman's Display if it has already been started in **LogData**.

### 2.2.2 Stop and start logging

Reset data recording by stopping the current session and starting a new one.

### 2.2.3 Stop Logging

Disconnect from NaviPac **Kernel** and close open logfiles.

**Note**: **Stop Logging** can also be done by clicking **Stop** on the Helmsman's Display if it was started from there.

### 2.2.4 Pause Logging

Temporarily pause the logging of data from NaviPac, but keep all logfiles open.

To continue again, click **Start Logging**.

### 2.2.5 Set up Logging

In the **Setup Logging Parameters** dialogue box, the user can set up what to log, where and how often to log it (**Recording Mode**), and select which information should be displayed in main window.

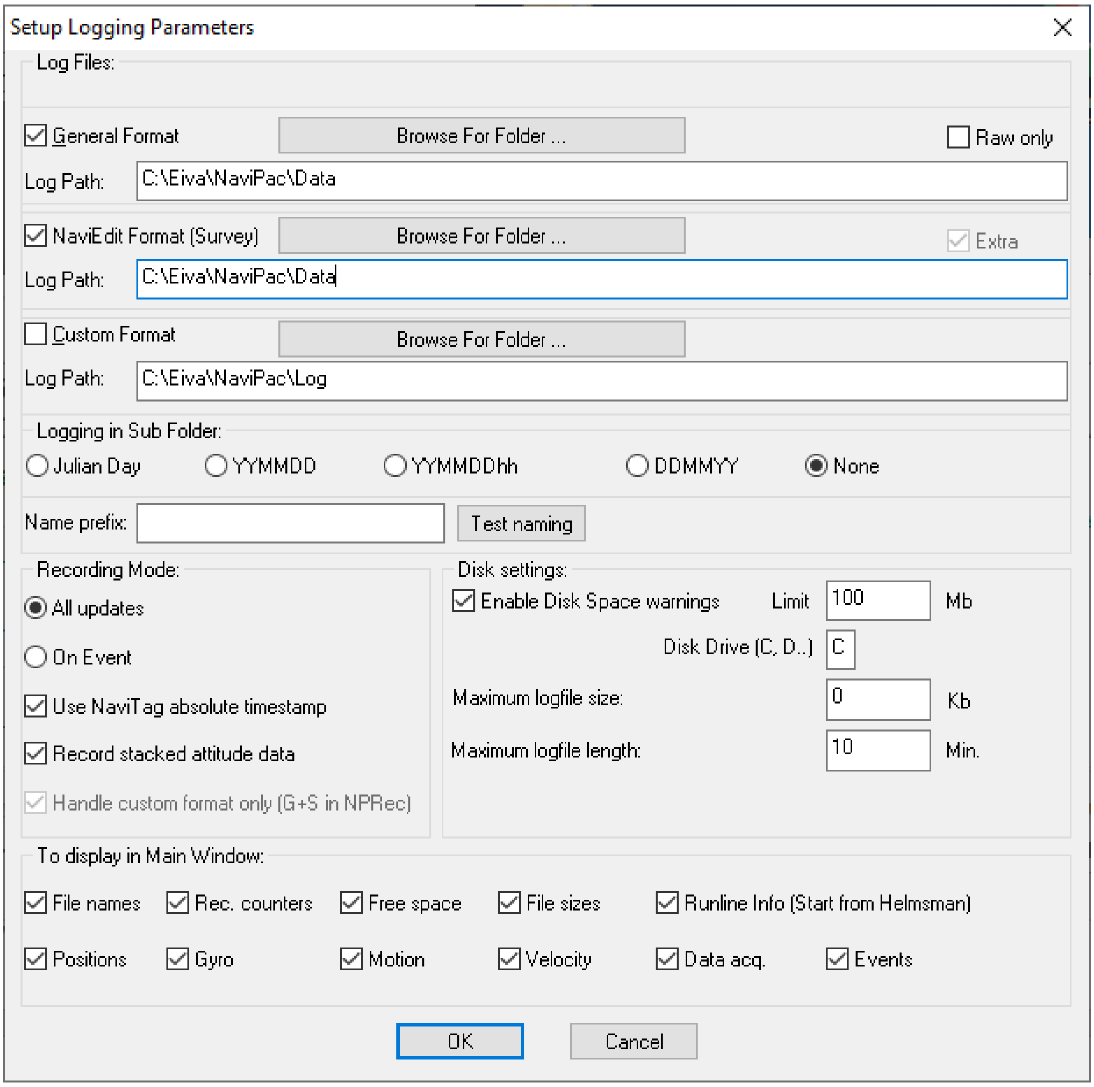


Figure 7 File, Setup Logging Parameters

#### 2.2.5.1 Log files

In the upper section (**Log Files**) you can specify the logging formats to log in and the directories and optional use of daily sub folders.

**Note**: You are not allowed to include a space in any part of the path name.

#### 2.2.5.2 Extra

Select this check box if extra information is needed in the survey format (eg **Logging of offset positions**).

#### 2.2.5.3 Raw only

Reduce general logging files to include only header and raw data strings (ie not storing processed data, as it can be found in survey format instead.)

#### 2.2.5.4 Sub folders and naming convention

Possibility to select 'Julian Day', YYMMDD, YYMMDD_HH,  DDMMYY or none for subfolders.

Files will always be named: YYYYMMDD_HHMMSS_<CGS>.NPD

- **Julian day**
  Current Julian day (eg logfile C:\Eiva\NaviPac\data\091\20090401_094411_G.NPD)
- **YYMMDD**
  Year, month, day (eg C:\Eiva\NaviPac\data\090331\20090401_094411_S.NPD)
- **DDMMYY**
  Day, month, year (eg C:\Eiva\NaviPac\data\ 010409\20090401_094411_S.NPD)
- **YYMMDDhh**
  Year(YY), month(MM), day(DD), hour(hh) (eg C:\Eiva\NaviPac\data\090401_09\20090401_095537_C.NPD)
- **None**
  NaviPac does not use sub folders. (eg. C:\Eiva\NaviPac\data\20090401_095537_C.NPD).

#### 2.2.5.5 Naming prefix

You may enter a prefix here that will be put in front of the official file name. Click **Test naming** to test the name.

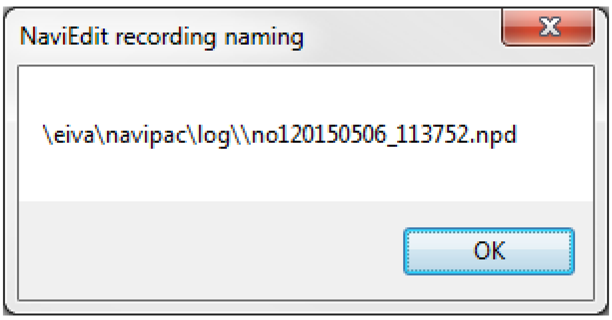



Figure 8 Display of sample file naming

#### 2.2.5.6 Recording Mode

Descriptions for the fields are as follows:

- **All updates**:
  Log on every cycle.
- **On Event**:
  Log when an event occurs in NaviPac.
- **Use NaviTag timestamps**:
  From NaviPac version 3.5 patch 5. Shall timestamp be based absolute time or local time?
  **Note**: If used with TimeBox, it is recommended that NaviTag timestamps are turned on.
- **Record stacked attitude data**:
  From NaviPac version 3.5 patch 8. Shall NaviPac record one or all attitude (gyro and motion) records per cycle?
  **Note**: NaviPac must be restarted to make this change.
- **Handle Custom format only**: If **NPRec** handles survey and general format (enabled in **NPRec**) then this check box is selected (read only), and **LogData** will only handle the custom format.

#### 2.2.5.7 Disk settings

Descriptions for the fields are as follows:

- **Enable disk space warnings**:
  If selected, a warning will be displayed in the main window when disk space on the disk drive specified is less than **Min. space left before warning** value.
- **Maximum log file size**:
  Enter max file size in kilobytes (max is 99999 kB) before **LogData** will switch to a new file. If set to 0, then no check is performed.
- **Maximum log file length**:
  Enter the maximum file size in time (minutes) before **LogData** will switch to a new file. If set to 0, then no check is performed.
- **Min. space left before warning**:
  Enter space on disk drive (in megabytes) before **LogData** should warn the user.

#### 2.2.5.8 To display in Main Window

If check box is selected, information for the item will be displayed in the main window, as seen in the figure below.

```
Mode: All,  0.00, Pos:    60, Gyro:     5, Att:     5, Speed:     5, Raw:      0, Depth:     5
Free Space: 2020.062012 Mbyte, 2068543.500000 Kbyte, 2118188544 bytes
XYZ rec. logf.: C:\Eiva\NaviPac\log\990323\990323S023.npd
P, D, 4004, 1999:03:23:14:28:35.798, X: 576201.45, Y:6224451.73, H:  -37.00, LAT: 56.15774, LON: 4.22678
G, 1999:03:23:14:28:35.694, Gyro: 0.0000, CMG:90.0000
A, 1999:03:23:14:28:35.714, R: 2.0400, P:-0.4500, H: 0.0000
V, 1999:03:23:14:28:36.638 S: 0.0000, SMG: 0.0022
D, 1999:03:23:14:28:35.921 ch:1, d:65.5700, b:-5.3371, d+b:60.2329
E2 1999:03:23:14:28:33.637 162
```

Figure 9 Status info in the log data window

- **File names**
  Display PATH and file name for each of the 3 formats if selected
- **Rec. counters**:
  Display line with total number of records received in since last program start
- **Free space**:
  Display space free on disk
- **Events**:
  Display event numbers
- **Positions**:
  Display P, Position Type, Time, E, N, Height, Latitude, Longitude
- **Gyro**:
  G, Time, Gyro, CMG
- **Attitude**:
  A, Time, Roll, Pitch, Heave
- **Velocity**:
  V, Time, Speed, SMG
- **Depth**:
  D, Time, channel, Depth, bathymetry, Depth + bathymetry

### 2.2.6 Save Settings

This will save all current settings for position / size / custom log format and logging setup. This function can also be reached from the toolbar.

### 2.2.7 Save As

Save the custom logging format in an external file for later use. The file is by default located in the **Data** folder and is an INI file type as seen in the figure below.

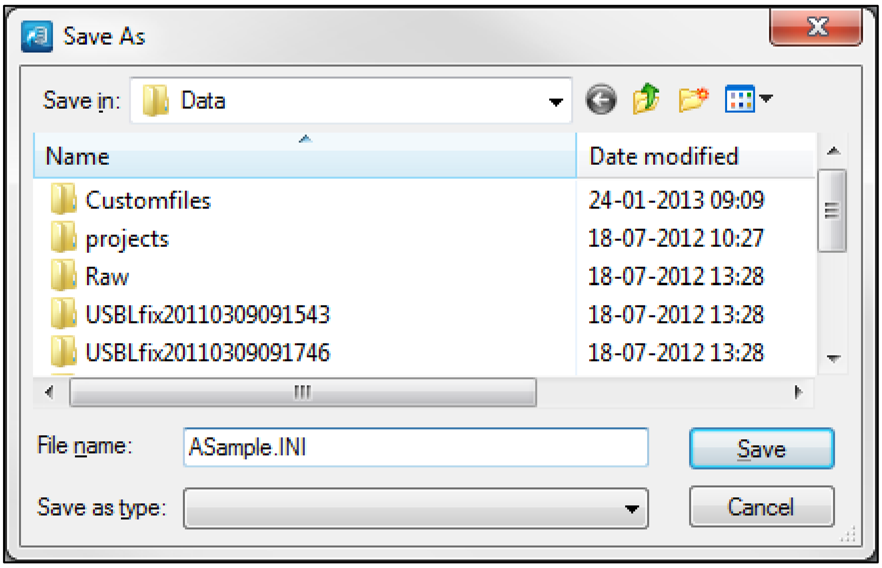
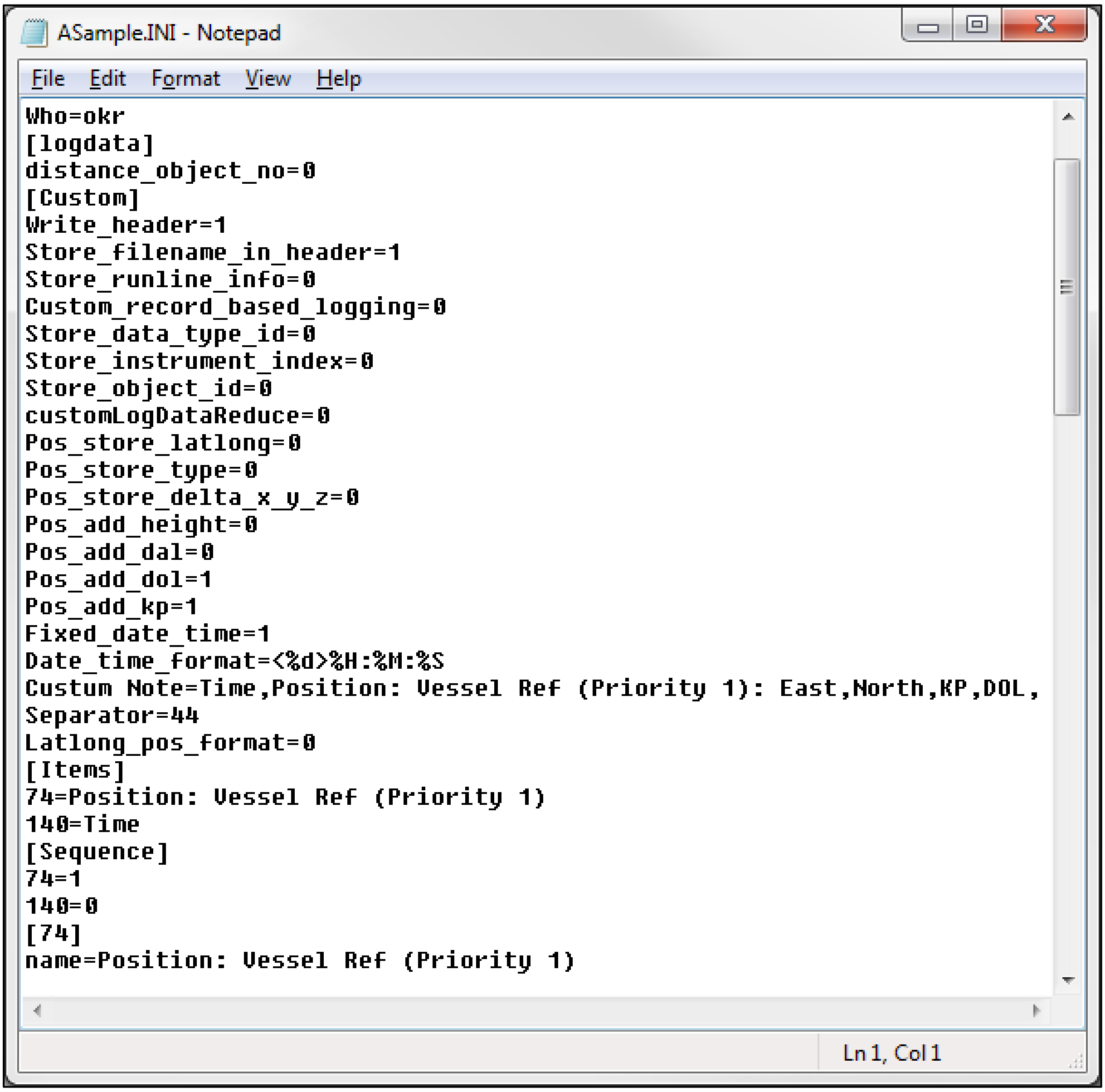


Figure 10 Standard file Save As dialogue box

**Note**: The file is context and contents sensitive, and you should not try to modify it manually.




```
Who=okr
[logdata]
distance_object_no=0
[Custom]
Write_header=1
Store_filename_in_header=1
Store_runline_info=0
Custom_record_based_logging=0
Store_data_type_id=0
Store_instrument_index=0
Store_object_id=0
customLogDataReduce=0
Pos_store_latlong=0
Pos_store_type=0
Pos_store_delta_x_y_z=0
Pos_add_height=0
Pos_add_dal=0
Pos_add_dol=1
Pos_add_kp=1
Fixed_date_time=1
Date_time_format=<%d>%H:%M:%S
Custum Note=Time,Position: Vessel Ref (Priority 1): East,North,KP,DOL,
Separator=44
Latlong_pos_format=0
[Items]
74=Position: Vessel Ref (Priority 1)
140=Time
[Sequence]
74=1
140=0
[74]
name=Position: Vessel Ref (Priority 1)
```

Figure 11 Resulting INI file

### 2.2.8 Load From

This menu item allows you to load a custom logging setup from earlier saved setup.

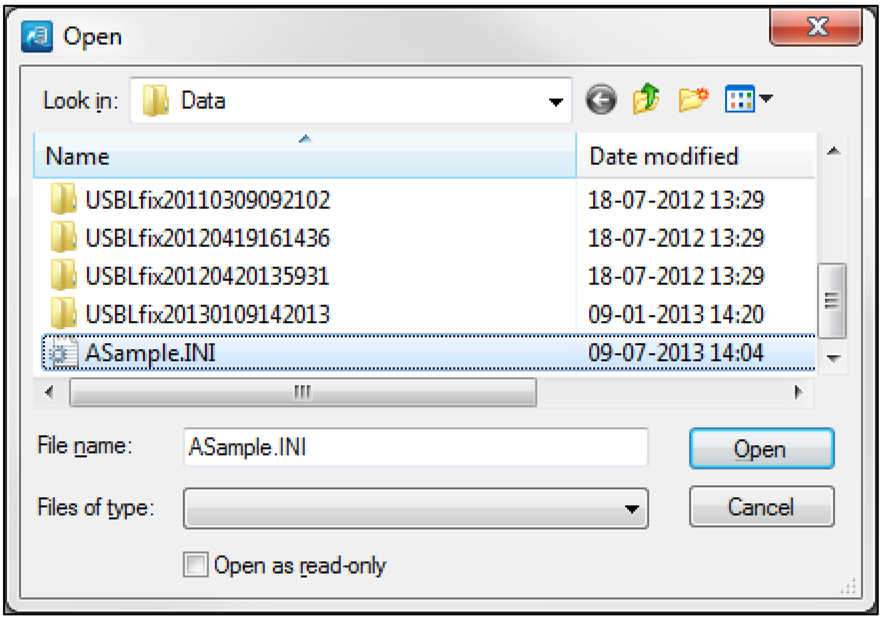



Figure 12 Select the definition file

Before loading the file, the program displays information about who created the file and when the last **Save** was performed, as seen below

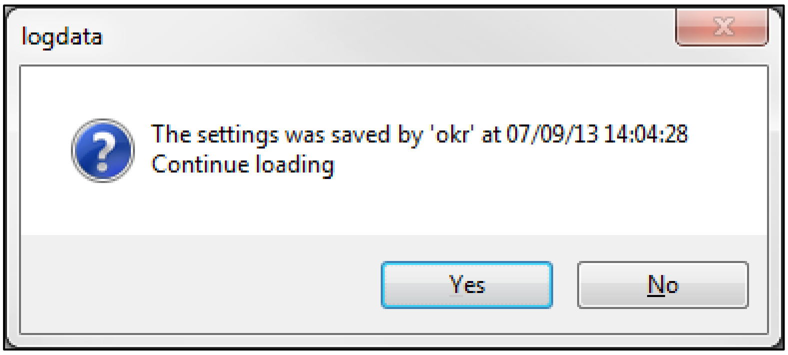



Figure 13 Status information about the saved definition

You may cancel by clicking **No**. If you accept by clicking **Yes** then the system resets the current custom logging format and loads the file. A short warning pops up, as seen below.

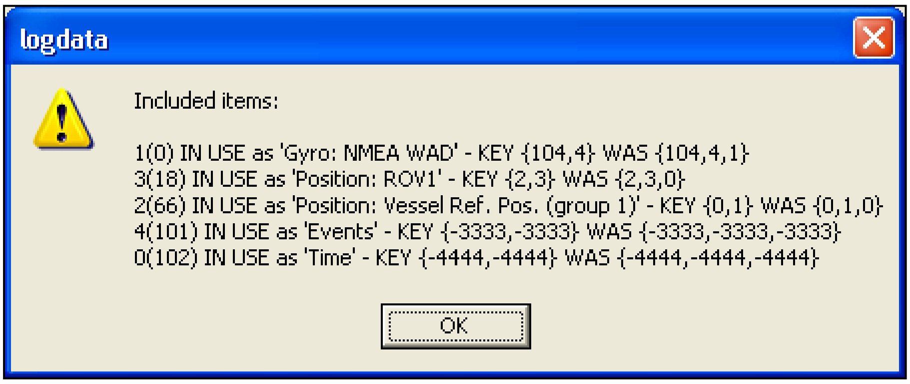


Figure 14 Error message in case of definition mismatch

The information window lists the units selected in the file, the sequence number, the name and references to setup.

If the external setup includes references to items that are not available in the current setup then a message pops up, as seen below.

Figure 15 Warning message

You should then check the setup using the normal custom logging setup.

**Note**: To make sure that **LogData** uses the new settings, please remember to **Save Settings**.

### 2.2.9 Exit

Stop logging (if started) and exit the program.

## 2.3 Edit menu

### 2.3.1 Edit Custom Log format

The **Custom LogData Format** can be specified in the dialogue box that appears when **Edit**: **Custom logging format** is selected.

#### 2.3.1.1 General Log Settings

The first tab holds general settings for the custom format, as seen below.

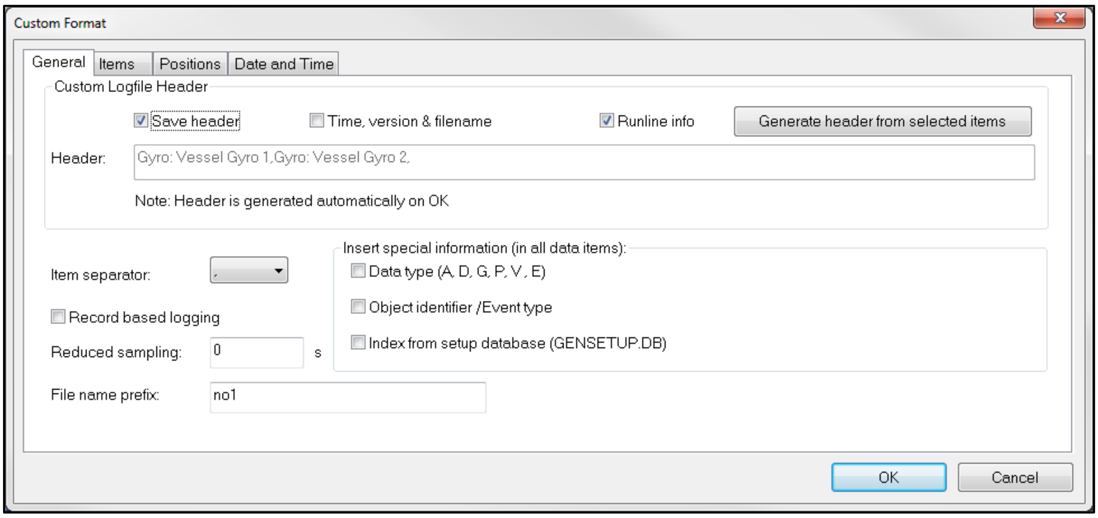



Figure 16 General tab for edit and custom logging format

#### 2.3.1.2 Custom Logfile Header

Fields that specifies what to log in the beginning of the custom defined logfiles.

##### 2.3.1.2.1 Save Header:

If selected, a header will be saved. The contents of the header are defined by the following fields/buttons (Time, etc).

##### 2.3.1.2.2 Time, version & filename:

Will include time, version and filename in the header.

**Example**:

1998:355:10:25:32.000 vers:2.2.0 file:C:\Eiva\NaviPac\log\981221\981221C000.NPD

##### 2.3.1.2.3 Auto generate header:

Insert all items from the included list and separate each by the selected separator.

**Example**:

Gyro: Anschutz NMEA 0183,Filtered vessel position: East,North,Height,

(If a gyro (`Anschutz NMEA`) and the Filtered vessel position was selected.)

As an alternative, the user can enter their own text that can be saved in the header of the logfile.

**Note**: If new items are added, and an auto-generated header is wanted, the dialogue box should be closed (by clicking **OK**) before the header can be auto-generated, as the items are first saved upon clicking **OK**.

#### 2.3.1.2.4 **Separator**:

An item separator is used to separate the items that are logged and it can be of following types:

- “, ” Comma (default)
- “ “ Space
- “; ” Semicolon
- “: ” Colon (be careful with date/time format)

#### 2.3.1.2.5 **Record based logging**:

The format of the logfile can be record-based (ie one record on each line) or line-based. Line-based logging (when record-based logging is not checked) means that all data items will stay on same line in logfile.

See sections 2.3.2.1 **Example 1 (record-based logging)** and **2.3.2.2 Example 2 (non-record-based logging)** for examples.

#### 2.3.1.2.6 Reduced Sampling:

The user can here determine if custom data will be logged more rarely than the NaviPac update. If you, for example, enter 5 seconds, then **LogData** will only write data samples once each 5 seconds.

### 2.3.1.3 Insert special information

The following general settings can be selected and thereby be logged for all items:

**Data type:**

Valid types are: P, G, V, A, D, E, C, R where P=Position, G=Gyro, V=Velocity, A=Attitudes, D=DataAcq (eg echo sounder channel), E=Events.

**Note**: C=Cycle info, R=RawData is not possible in custom format.

**Object Id:**

The object number/offset number (vessel = 0). Please see **Object Monitor** manual for further details on the ID field.

Note: **Object Id** cannot be logged for **Events** (E).

**Index in GENSETUP.DB/Event method:**

Instrument index or Event method, if an Event is logged.

#### 2.3.1.4 Items to be logged page

The **Available** list holds all available data items that can be logged. Only items that are moved to the **Selected** list (on the right of the below figure) will be logged. The sequence of the items goes from top to bottom in the **Selected** list. This sequence is only used when **Record based logging** is not selected.

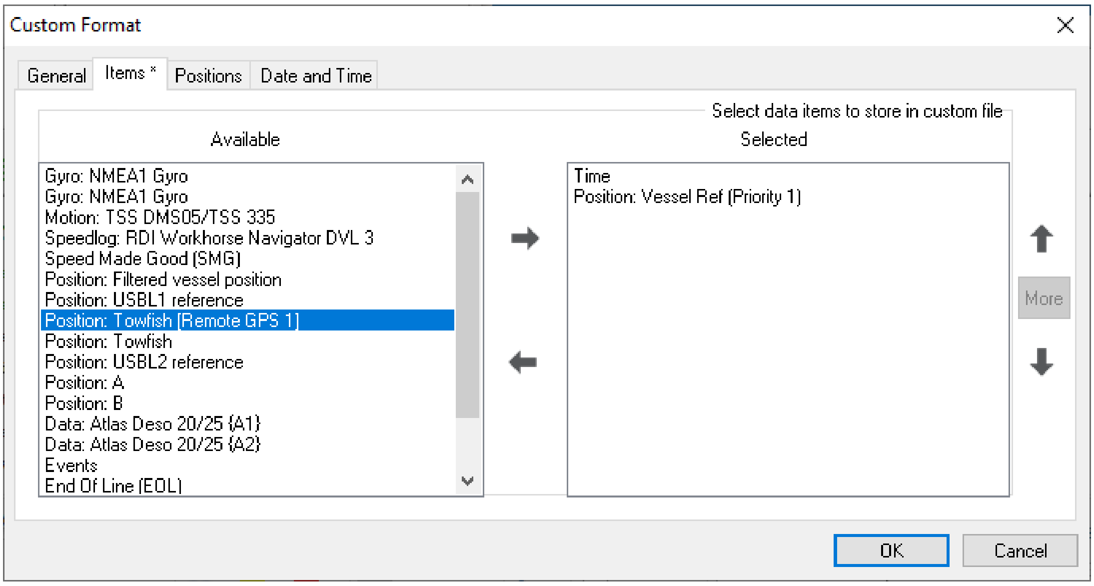


Figure 17 Item selection

**Note**: You are able to select multiple items in the **Available** list at a time by using CTRL + click.

The following data items/records can be selected (moved to **Selected** and thereby logged):

- **Time** (default: HH:MM:SS.sss)
  If **Custom date/time** is selected, the format can be user-defined. Use **Help** (in Date & Time) to specify another format. If **Fixed** is selected, the format will be HH:MM:SS.sss for all items.
- **Height** in positions records
- **Gyro** (**Note**: Not CMG)
- **Attitude info**
  Roll, Pitch, Heave
- **Speed info**
  Speed, SMG
- **All dynamic, fixed, filtered positions**
- **Ref. Position** (group 1-5): X, Y
  Only actual selected number of reference groups.
- **Depth** from each echo sounder channel
- **User-defined offsets** (X,Y, Height)
- **Events** (event number)

**Note**: ROVs are examples of dynamic positions. All offset positions defined in NaviPac setup/online will be selectable.

##### 2.3.1.4.1 Up Button

Moves the selected item one level up.

##### 2.3.1.4.2 Down Button

Moves the selected item one level down.

##### 2.3.1.4.3 More Button

Allows the operator to set item-specific flags.

##### 2.3.1.4.4 Date & Time on items

Date and time on individual items can be selected by selecting an item and clicking **More**, or by double-clicking the item in the **Selected** list.

An **Item Settings** dialogue box will pop up:

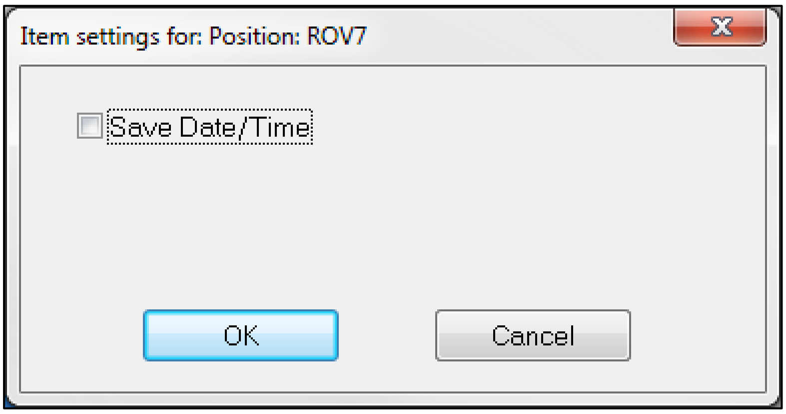


Figure 18 Save Date/Time for an item that is to be logged

#### 2.3.1.4.5 Echo sounder channel: More

Double-click in **Selected** list on an echo sounder channel (depths). A dialogue box pops up:

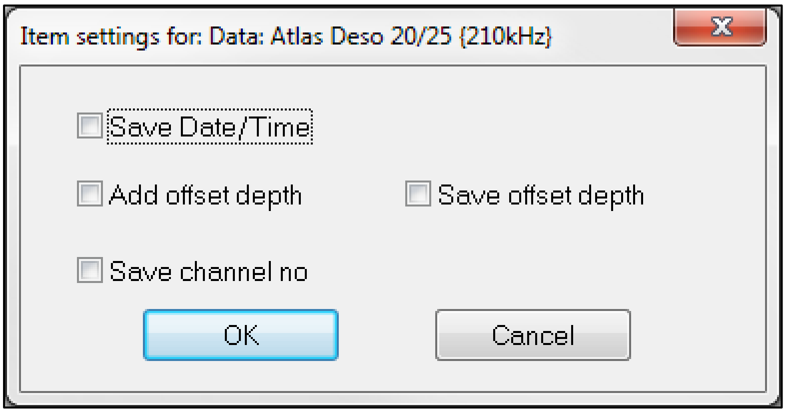


Figure 19 Settings for echo sounder and data acquisition channels

This enables the addition of the **offset depth** to the echo sounder depth while logging data.

**Note**: Adding the **offset depth** is normally needed to get the 'true' depth of the seabed. It can also be logged as a separate item.

The channel number can also be logged.

**Note**: Date/Time information on individual items is disabled by default. To save these settings, use **Save Settings** in the **File** menu.

#### 2.3.1.4.6 Gyro: More

When double-clicking on a Gyro in the **Selected** list, the following dialogue box pops up:

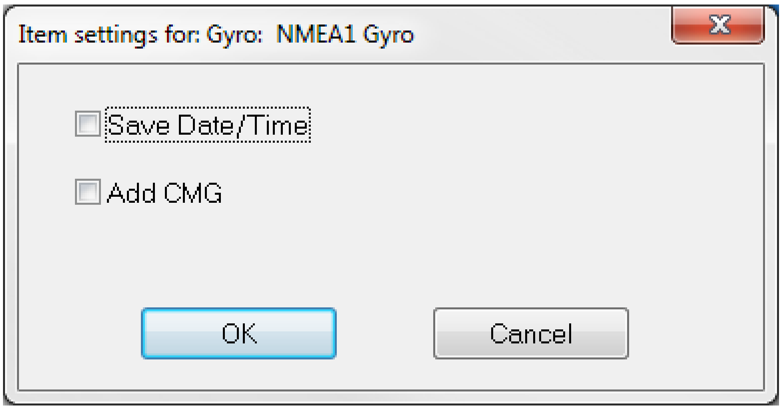



Figure 20 Gyro – More

This enables saving the CMG (Course Made Good) as a second field after the gyro. The default measurement is degrees.

**Note**: To save these settings, use the general **Save Settings** in the **File** menu.

#### 2.3.1.5 Positions tab

The **Positions** tab allows you to specify additional position information parameters. If you select a parameter to be logged it will apply to all position logged.

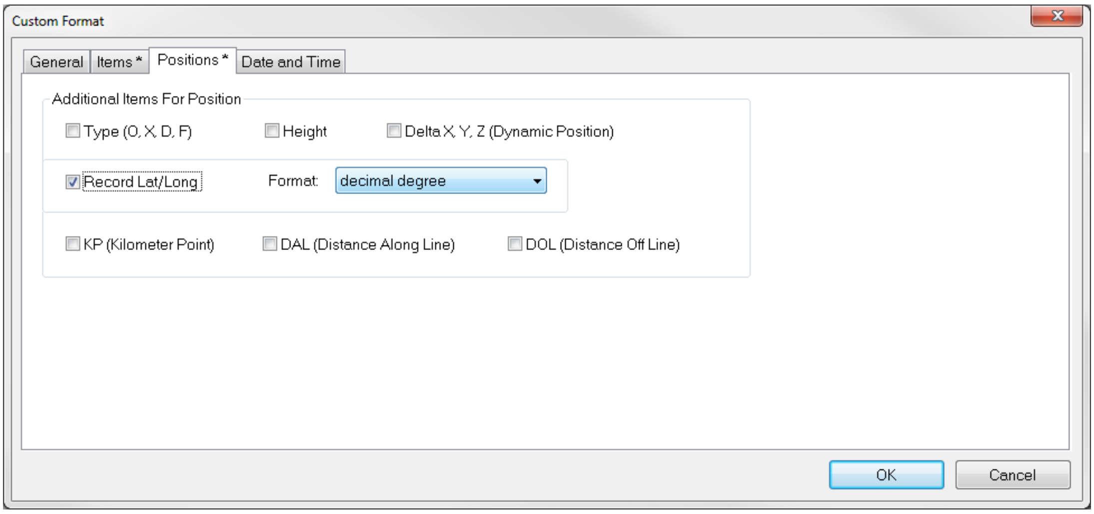



Figure 21 Positions tab

- **Type**: ordinary positions (**O**), fixed positions = user-defined offsets (**X**), dynamic positions (**D**) (eg ROV) filtered positions (**F**)
- **Height**: Operator selects whether NaviPac includes height as part of position or not?

- **Delta XYZ**: Whether to include the relative position information (eg. dX/dY/dZ from USBL)
- **Pos. as Lat/long**: Log Latitude, Longitude in positions records. The format can be decimal degree, Radians, DDD°MMM"SS.SSSSS, DDD°MMM.MMMMMM
- **KP** (Kilometre Point)
- **DAL** (Distance Along Line)
- **DOL** (Distance Of Line)

**Note**: The last 3 items (KP, DAL, DOL) will only be logged if logging is started from Helmsman's Display (eg a runline is loaded/created and line is red).

#### 2.3.1.6 Date and Time tab

The **Date and Time** tab allows you to specify how time is logged, if **Time** is selected in **Items to be logged**.

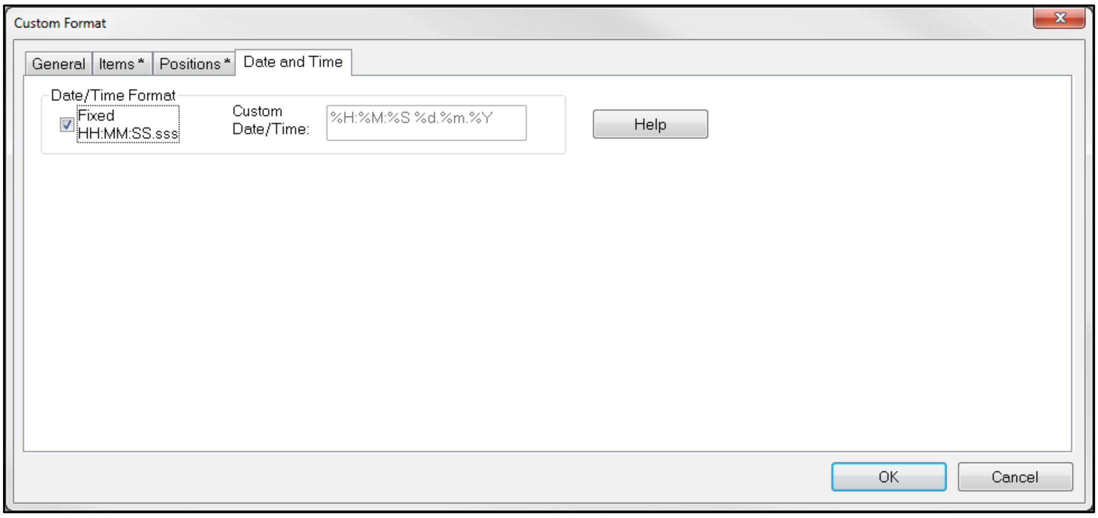


Figure 22 Date and Time tab

**Time** is by default selected to the fixed format. If **Custom Date/Time** is selected, then you can define the format.

The format argument consists of one or more codes; as in C/C++ **printf**, the formatting codes are preceded by a percent sign (%). Characters that do not begin with % are copied unchanged to logfile. The formatting codes for date and time in custom logfiles are listed below:

%a Abbreviated weekday name
%A Full weekday name
%b Abbreviated month name
%B Full month name
%c Date and time representation appropriate for locale

%d Day of month as decimal number (01 - 31)
%H Hour in 24-hour format (00 - 23)
%I Hour in 12-hour format (01 - 12)
%j Day of year as decimal number (001 - 366): (=Julian day)
%m Month as decimal number (01 - 12)
%M Minute as decimal number (00 - 59)
%p Current locale's am./pm indicator for 12-hour clock
%S Second as decimal number (00 - 59)
%U Week of year as decimal number, with Sunday as first day of week (00 - 51)
%w Weekday as decimal number (0 - 6; Sunday is 0)
%W Week of year as decimal number, with Monday as first day of week (00 - 51)
%x Date representation for current locale
%X Time representation for current locale
%y Year without century, as decimal number (00 - 99)
%Y Year with century, as decimal number
%z, %Z Time-zone name or abbreviation; no characters if time zone is unknown
%% Percent sign

#### 2.3.1.6.1 Example

Specifying: %H:%M:%S %d.%m.%Y

Will result in the following appearance in the logfile:

10:06:01 12.01.1998

**Note**: Clcik **Help** in dialogue box to specify another format.

If **Fixed** is selected, the format will be HH:MM:SS.sss for all items.

### 2.3.2 Examples of Custom Log format

In the following, it is assumed that the user has selected record-based time, x,y, lat, lon, events and a comma separator:

#### 2.3.2.1 Example 1 (record-based logging) with record type ID

The following items have been selected in the **Items** tab:

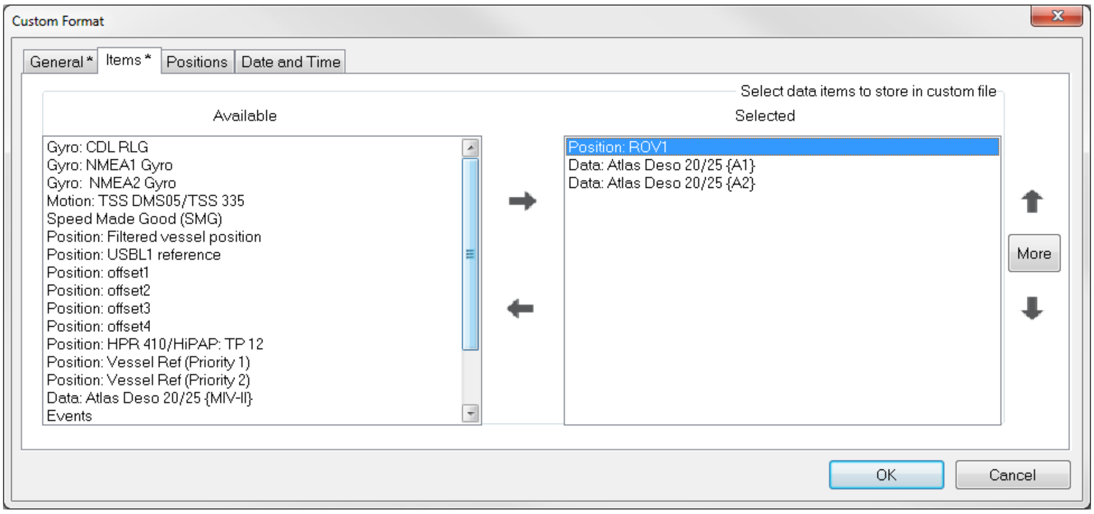


Figure 23 Items selected for record-based logging

In the General tab, the following is selected:

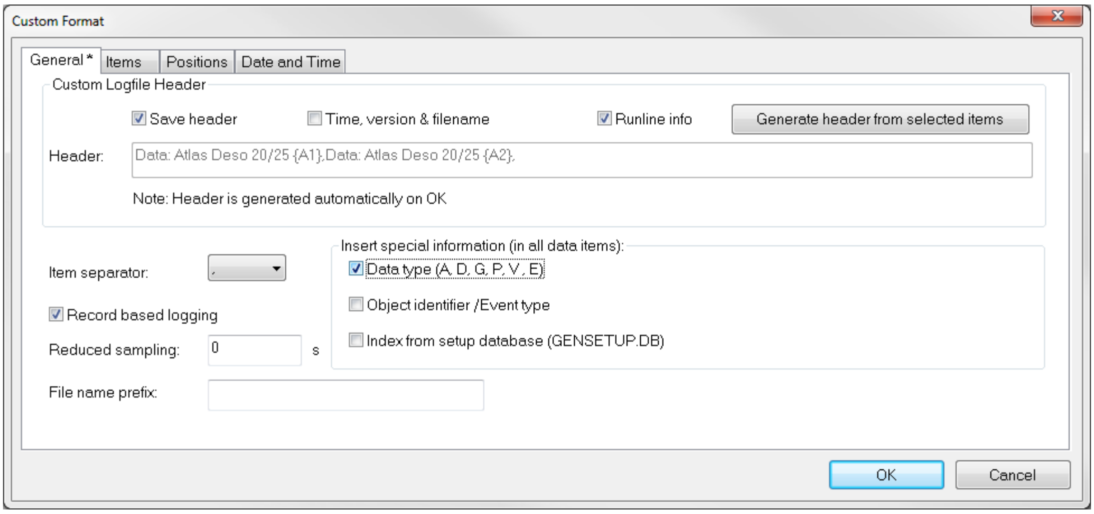


Figure 24 General setup for sample recording

The resulting file will look the following:

1998:355:12:04:18.000 vers:2.2.0 file:C:\Eiva\NaviPac\log\981221\981221C004.npd Time,NaviSound 2000:1: East,North,Height,NaviSound 2000: NaviSound 2000:1,Events,

T,12:04:18.794

P, 576267.6104,6225456.7066, -9.8663

D, 45.8163

E, 30

T,12:04:19.794

P, 576267.6416,6225456.9101, -9.8661

D,  45.9161
T,12:04:20.795
P, 576267.6734,6225457.0947, -9.8659
D,  46.0259
E,    31
T,12:04:21.795
P, 576267.7158,6225457.2978, -9.8658
D,  46.1258
E,    32

### 2.3.2.2 Example 2 (non-record-based logging)

In this example, the same **Items** are selected as the example above. The **General** tab has the following selected:

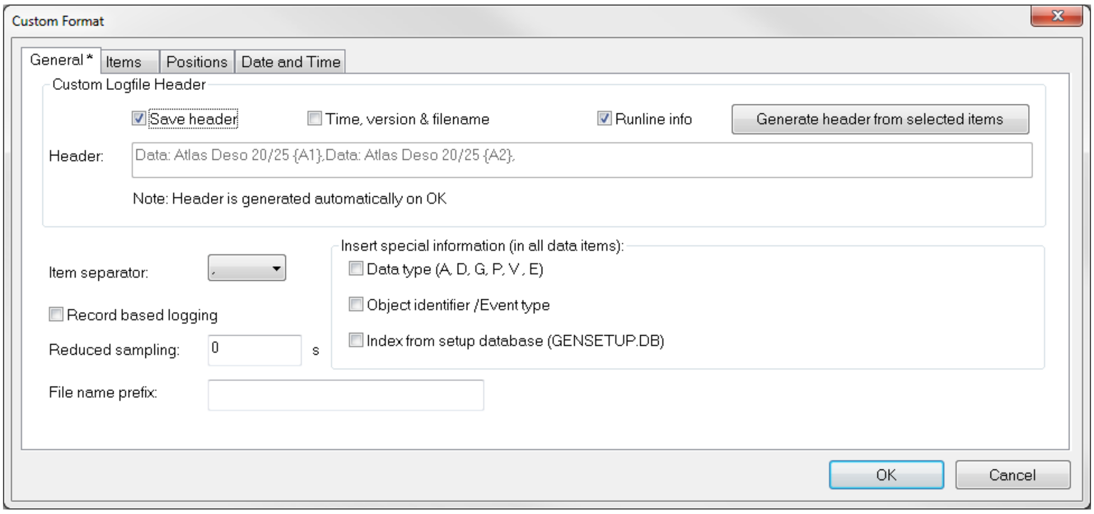


Figure 25 General settings for non-record based recording

The resulting file will look like this:

1998:355:11:59:11.000 vers:2.2.0 file:C:\Eiva\NaviPac\log\981221\981221C003.npd Time,NaviSound 2000:1: East,North,Height,NaviSound 2000: NaviSound 2000:1,Events,
11:59:11.735,  576256.8172,6225396.2787, -9.8831,   46.3231,     23
11:59:12.736,  576256.8398,6225396.4827, -9.8830,   46.4230,     25
11:59:13.736,  576256.8741,6225396.6703, -9.8830,   46.5230,     25
11:59:14.735,  576256.8967,6225396.8743, -9.8829,   46.6329,     25
…
11:59:19.736,  576257.0333,6225397.8632, -9.8827,   47.1427,     28

## 2.4 View menu

### 2.4.1 View: Custom log file

This menu displays the current custom specified logfile by calling up a user-defined editor program with the ASCII file (eg Notepad can be specified in the NaviPac.INI file in $WINHOME: by the line: *LOG_FILE_VIEWER_PROGRAM=notepad)*.

See section 3.2 Setup of this manual for more details.

### 2.4.2 View: Logging History

For troubleshooting; provides access to recorded trace logfile. The logging history is often good to hand over to EIVA Support in case of technical queries.

## 2.5 Options menu

### 2.5.1 Save logging history to file

Use this option if information from the **Start/Pause/Stop** actions and the logfiles created should be logged.

**Note**: This function can save a lot of time after a survey, so its use is highly recommended.

The file is placed in: <Install drive>: \EIVA\NaviPac\Log\History\ logdata.log.

**Example of history file**:

Started Logging at time: Thursday, October 05, 2000 12:32:46 - by USER
recording_mode: Every cycle, Rate: -1, Distance=-1.000000
Naviline logging enabled. File=C:\Eiva\NaviPac\log\001005\001005S008.npd
Custom logging enabled. File=C:\Eiva\NaviPac\log\001005\001005C037.npd
StartLogging; GenFile=, NEFile=001005S008.npd,CustFile=001005C037.npd
Start Pos: X:500779.764002, Y:6004424.412308
Stop Pos: X:500780.240913, Y:6004427.170439
Stopped Logging at time: Thursday, October 05, 2000 12:32:54 - by USER
Statistic: Vel: 8, Dataacq: 8, Raw: 0, Motion: 0, Gyro: 8, Pos: 72, Cycle: 8, Events: 4

Started Logging at time: Thursday, October 05, 2000 12:33:04 - by USER
recording_mode: Every cycle, Rate: -1, Distance=-1.000000
Naviline logging enabled. File=C:\Eiva\NaviPac\log\001005\001005S009.npd
Custom logging enabled. File=C:\Eiva\NaviPac\log\001005\001005C038.npd
StartLogging; GenFile=, NEFile=001005S009.npd,CustFile=001005C038.npd
Start Pos: X:500781.076380, Y:6004431.910910

Paused Logging at time: Thursday, October 05, 2000 12:33:04

### 2.5.2 Reset logging history

Clear the history file.

**Note**: Remember to make a copy of the history file if it is wanted for later use.

## 2.6 Help menu

### 2.6.1 Logging formats

Display Online help for **LogData** program (this document) using an HTML viewer (like Microsoft Internet Explorer).

### 2.6.2 About logdata

Shows current program version, icon and copyright note.

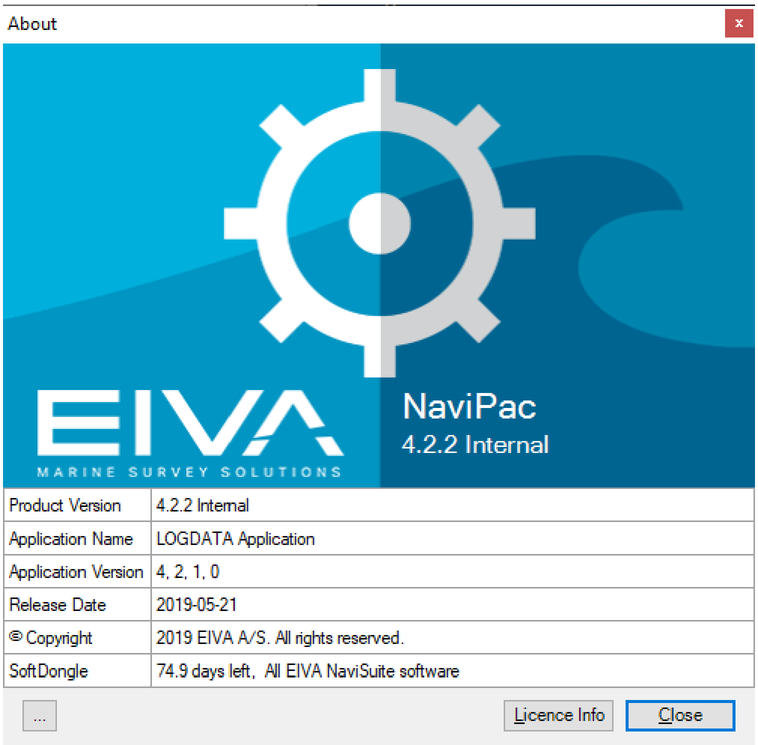



Figure 26 The About dialogue box

# 3 Start, Stop and Setup of LogData

The **LogData** program is normally started from the NaviPac Online program.

## 3.1 Start and stop logging to logfiles.

The logging is started when the users select **File: Start logging** or **File: Stop and Start logging**. The last will first stop the previous logging - if any. This can also be done from the toolbar.

To stop all logging, select: **File: Stop logging**.

In **File: Setup Logging** you specify which logging formats to use.

## 3.2 Setup

**LogData** can be configured in the NaviPac.INI system file in the Setup directory. The following entries are used:

```
[common]
LOGDATA_SOCKETPORT=4885
; tcpip server addr. for data to online, qc, attitude, gpsstatus, objpos, basepos
SERVER_DATA_ADDR_N1=192
SERVER_DATA_ADDR_N2=6
SERVER_DATA_ADDR_N3=21
SERVER_DATA_ADDR_N4=5

;DATA Logging settings
[logdata]
; set WARMSTART=0 if should not be started automatic
WARMSTART=0
TRACEFILE=logdata.log
;enable trace by setting TRACE=1, disable by TRACE=0
TRACE=1
HTML_HELP_FILE=c:\eiva\navipac\help\logdata\logdata.htm
HTML_DATE_FORMAT_HELP_FILE=c:\eiva\navipac\help\logdata\datetimeformats.htm

LOG_IN_NAVILINE_FORMAT=0
NAVILINE_LOG_PATH=C:\Eiva\NaviPac\log
LOG_IN_GENEREL_FORMAT=0
GENEREL_LOG_PATH=C:\Eiva\NaviPac\log
LOG_IN_CUSTOM_FORMAT=1
CUSTOM_LOG_PATH=C:\Eiva\NaviPac\log
```

```
;LOGFILE_PREFIX: 0:JJJ, 1:YYMMDD, 2:_DDMMYY
LOGFILE_PREFIX=1
GEN_LOG_FILE_POST_NAME= G000.npd
NAVILINE_LOG_FILE_POST_NAME= N000.npd
CUSTOM_LOG_FILE_POST_NAME= C000.npd

DISPLAY_RECORD_COUNTER=1
DISPLAY_GYRO_RECORDS=1
DISPLAY_EVENT_RECORDS=1
DISPLAY_ATTITUDE_RECORDS=1
DISPLAY_POSITION_RECORDS=1
DISPLAY_VELOCITY_RECORDS=1
DISPLAY_DEPTH_RECORDS=1
DISPLAY_LOGFILE_NAMES=1
DISPLAY_SPACE_LEFT=1
ENABLE_WARNINGS=1
; warning if space less than 10 Mbyte
MIN_SPACE_IN_MB_BEFORE_WARNING=100
SPACE_LEFT_DISK_DRIVE=C
;LOG_FILE_VIEWER_PROGRAM=Write
LOG_FILE_VIEWER_PROGRAM=notepad
;MODES 0: log all records, 1: time - e.g. log when LOGGING_RATE_IN_SEC passed, 2: event
LOGGING_MODE=0
LOGGING_RATE_IN_SEC=5
LOG_POS_DISTANCE=0.000000
; 1: Em1000 logging The em1000 logging should be started, 0: do no log
LOG_EM1000_DATA=0
NP_LOGGING_STARTED=0
NP_CURRENT_LOGFILE=C:\Eiva\NaviPac\log\981221\981221C004.npd
StartAlways=0                    // set to 1 to start loggig when NaviPac starts
```

# 4 Log Data Formats

As discussed earlier, there exist 3 different log format possibilities:

1. A general (hard coded) format that will contain all information including setup of instruments, raw instrument data, events.
2. A hard coded format to be used by a NaviPac interpreter (NaviEdit version). Single-beam echo sounder data can be logged and read into NaviEdit for data editing.Export to (for example) XYZ files for charting. This is the preferred format for survey data logging.
3. A customizable format, in which the user can select which information to log into a comma-separated ASCII file.

These formats are further described in this chapter.

## 4.1 General format

This logging format will hold all information it is possible to log in the NaviPac system

**Note**: We recommend using the Custom or Survey logging format for post processing purposes, as it will be much easier to interpret.

The General format uses much more disk space than the other formats.

The Custom format is ideal for spreadsheets etc.

The General format is, on the other hand, good for recalculation and error situations since it holds all instrument settings, raw data and events etc.

The data format is contained in a plain ASCII file, therefore in DOS each line is terminated by <CR><LF>.

The file consist of a general header describing the current setup, followed by different Navigation records (eg raw positions from GPS, from offsets, ranges from Navigation. stations, attitudes, SMG, CMG etc)

**All** items are always separated by a space in this general format (ASCII HEX 20).

The data format is structured in records of data items.

The first character of each record defines the type of record:

H (Header record),
E (Event record)
C (Cycle Record)
P (Position record),
R (Raw data record)
D (Data acquisition record, eg depths from echo-sounders)
A (Attitude record: RPH)
G (Gyro record)
V (Velocity record)

All records will include time and date information in the start of the record.

The second is sometimes used as the (sub) type of the record:

0, 1, ...500

If a record contains more than one text line, it is started by:

<Record type character><subtype character><new-line>

and terminated by:

***/***< Record type character>< subtype character><new-line>

See section 4.1.1.1. **Example of the Logfile Header record** for an example of such a large record.

Data items which may not exist are set to the default values.

The maximum length of one text line in the ASCII file is 1024 characters.

Longitude, Latitude have the format: DDD.mmmmm = decimal degrees. This format has been chosen to avoid non-portable ASCII characters (Windows->Unix).

### 4.1.1 Header record type (multiple lines)

The header file consists of general setup information created by the NaviPac Setup program. This information can also be displayed (and printed) separately from this program.

After the general setup, the Online setup information is included (eg information for the LOPs currently selected in the Online program). The Setup program also generates this information.

**Note**: It is therefore necessary that the Setup program always generates new ASCII files with General and Online setup each time the user make changes to the setup as it is used by **LogData**.

The header consists of information like:

Start of header id = 0 (H0)
Logging format version id
-------------------------------------------------------
Date & Time for **General** set-up database creation
User that created/modified the database.
If restart of NaviPac is enabled or disabled
The number of Navigation systems.
The number of Navigation stations
The number of Navigation instruments
Dongle information
-------------------------------------------------------
The current ellipsoid, projection and
Datum shift parameters
Data scale (unit)
-------------------------------------------------------
Environmental data (gravity, sound velocity)
Survey information (vessel, client, area and job)
Height/geoidal information (GPS height to user height)
Global parameters (info from Setup global and underwater parameters)
Time zone information
-------------------------------------------------------
The names of defined navigation systems
-------------------------------------------------------
Parameters for each defined Navigation station (eg easting, northing, height)
-------------------------------------------------------
Setup for each defined Navigation Instrument (grouped into: Surface navigation, Gyro, Speedlog, Motion (rph), Dynamic position, Data output, Data acquisition)
Mode, port number, baudrate, databits, stopbits, offset: X Y Z - e.g. [Sim] Port 05 IO  9600 N 8  XYZ: -0.03  3.03
-------------------------------------------------------
Date for **Online** setup database creation

Number of priorities, number of LOPs, number of additional LOPs, number of offsets (objects)
Time controlled by (operator/GPS system)
Reference_prio_grp
Estimated position (X,Y)
Selected LOPs including: weight, sigma, online c-o, priority group, set-up co1, set-up c02
Selected Objects including: object name, type, instrument index, and TPII number.
End of Header id (/H0)

**Note**: This format is changing as NaviPac evolves, and since it is in readable ASCII format, the above description is only a snapshot.

#### 4.1.1.1 Example of the Logfile Header record: H0

DB ascii dump: C:\EIVA\DB\Gensetup.DB
Program: NPConfig
Date: 18.09.2011 14:45
User: okr
UserId: 0
ProgId: 3.8.0.0
Restart mode: Enabled
Editing mode: Saved
Systems: 3
Stations: 3
Instruments: 12
----------------------------------------------------------
Dongle info:
Dongle 544137 - Full + TMS + 3DMon
----------------------------------------------------------
Ellipsoid : WGS 84
Inv. flat.: 298.2572235630
Semi major: 6378137.0000
Projection: UTM (north)
Proj. type: 05
Org. scale: 0.999600000000
1.Parallel: 033°00'0.0000"
2.Parallel: 045°00'0.0000"
Longitude: 009°00'0.0000"
Latitude : 000°00'0.0000"
Easting : 500000.0000
Northing : 0.0000
UTMzone : 32
----------------------------------------------------------
Datum Shift: None
Method: None

Tx : 0.000000 m
Ty : 0.000000 m
Tz : 0.000000 m
Rx : 0.0000000000 °
Ry : 0.0000000000 °
Rz : 0.0000000000 °
PPM : 0.000000
-----------------------------------------------------------
Selected data scale:
Position: metric scale 1.00000000 Unit m
Depth: metric scale 1.00000000 Unit m
Logging: Pos 0 Depth 0
-----------------------------------------------------------
Environmental data:
Gravity : 9.82000 m/s^2
Pressure Surf : 1.00000 Atm
Density Water : 1.00000 kg/dm^3
Sound velocity: 1499.48 m/s
Time zone info: <Romance Standard Time> [Romance Daylight Time] Zone: -3600 DL 1
-----------------------------------------------------------
Survey information:
Area : Århus
Client : EIVA
Job : Test123{OKR}
Vessel : MS-EIVA
IMCA Id : NP
-----------------------------------------------------------
Height/geoidal data:
Geoidal red. : Manual
GPS H ->Offset: No
-----------------------------------------------------------
Global parameters:
Deskewing : No Filter 10.00
Cycle freq. : 1.00
CMG/SMG Filt. : 10.00 30.00 51.00 46.00 56.00
GPS setting : Use all
USBL Setting : Stacked Surface = No
: Use time from HiPAP = Yes
: Hold time = 5.00 sec
: Tolerance = 100.00 %
-----------------------------------------------------------
Navigation Systems:
1: Topcon GPS 1 - Topcon GPS 1
2: Topcon GPS 2 - Topcon GPS 2
3: Topcon Total Station 1 - Topcon Total Station 1

----------------------------------------------------------
Navigation Stations:
 0:Topcon GPS 1-
  avail: OK, COe: 0.00, COn: 0.00, Use age/time: OK, Fixed age: 0.00, GeoH: 0.00, DatumS: None (0)
 1:Topcon GPS 2-
  avail: OK, COe: 0.00, COn: 0.00, Use age/time: OK, Fixed age: 0.00, GeoH: 0.00, DatumS: None (0)
 2:Topcon Total Station 1-NEW2
  avail: OK Position E: 500000.00, N: 6300000.00, H: 12.00
----------------------------------------------------------
Navigation Instruments:

----------- Surface Navigation ------------
 1: Topcon GPS 1 type: 31, seq: 0, Mode [Sim] Port 03 IO 115200 N 8 1 XYZ: 0.000 0.000 0.000
   Command to GPS: FALSE, Use PPS: FALSE, Use ZDA: FALSE, command file:
 2: Topcon GPS 2 type: 32, seq: 0, Mode [Sim] Port 02 IO 115200 N 8 1 XYZ: 0.000 0.000 0.000
   Command to GPS: FALSE, Use PPS: FALSE, Use ZDA: FALSE, command file:
 3: Topcon Total Station 1 type: 54, seq: 0, Mode [Sim] Port 01 IO 9600 N 8 1 XYZ: 0.000 0.000 0.000

----------- Gyro ------------
 4: NMEA1 Gyro type: 104, seq: 0, Mode [Sim] Port 15 IO 9600 N 8 1 XYZ: 0.000 0.000 0.000
   CO: 0.000, Time Slice: 0.000, Offset: Vessel, obj.no: 0 Delay: 0.000, OBJ: 0,18,20,-1,-1
 5: Anschutz NMEA 0183 type: 153, seq: 0, Mode [Sim] Port 14 IO 9600 N 8 1 XYZ: 0.000 0.000 0.000
   CO: 0.000, Time Slice: 0.000, Offset: Antenna2, obj.no: 440 Delay: 0.000, OBJ: 440,-1,-1,-1,-1

----------- Motion sensor ------------
 6: CDL RLG type: 228, seq: 0, Mode [Sim] Port 12 IO 9600 N 8 1 XYZ: 0.000 0.000 0.000
   Roll CO: 0.000, Pitch CO: 0.000, Heave CO: 0.000, Time Slice: 0.000, Offset:Vessel, obj.no: 0 Delay: 0.000, OBJ: 0,18,20,-1,-1

----------- Dynamic positioning ------------
 7: HPR 410/HiPAP type: 496, seq: 0, Mode [Sim], IO Type: UDP/IP, port: 0000, IP addr: 000.000.000.000 XYZ: 0.000 0.000 0.000
   USBL Settings: HT: 12 sec GRP Add angles: 0.00 0.00 0.00 (°) RangeScale 1.000000
     RECEIVER-02 , obj.no 18, type , id 1, XYZ 0.000 0.000 0.000
     ROV8 , obj.no 20, type , id 2, XYZ 0.000 0.000 0.000
 8: Comb. dynamic pos 1 type: 440, seq: 0, Mode [Cal] Port 01 IO 9600 N 8 1 XYZ: 0.000 0.000 0.000
  Comb. dynamic pos 1 , obj.no 440
  Based on 2 primary objects:
     Basis object (0)= RECEIVER-02 , X: 0.000, Y:4.000, Z: 1.000
     Basis object (1)= ROV8 , X: 0.000, Y:-4.000, Z: -1.000

----------- Data Output ------------
 9: Eiva runline control type: 649, seq: 0, Mode [On ], IO Type: UDP/IP, port: 6005, IP addr: 127.000.000.001 XYZ: 0.000 0.000 0.000
   Offset=Vessel, obj.no: 0 Pos/Depth Scaling: No

10: Heave correction and GPS height  type: 621, seq: 0, Mode [On ], IO Type: UDP/IP, port: 6007, IP addr: 127.000.000.001 XYZ: 0.000 0.000 0.000
    Offset=Vessel, obj.no:   0 Pos/Depth Scaling: No
11: Position (Exp.) to NaviScan      type: 671, seq: 0, Mode [On ], IO Type: UDP/IP, port: 5555, IP addr: 127.000.000.001 XYZ: 0.000 0.000 0.000
    Offset=Vessel, obj.no:   0 Pos/Depth Scaling: No

----------- User Defined Offsets ------------
12: O01                         type: 801, Mode [Cal], XYZ: 0.000 0.000 0.000
    name=O01                         , obj.no:   0, type=Gyro
----------------------------------------------------------
This information is saved in 'C:\EIVA\log\current.setup.txt' in ASCII format
----------------------------------------------------------

### 4.1.2 E (Event record)

Events can be StartOfLine (2), StopOfLine(3), StopStartLine(4).

Timeevent(5), ExternalEvent(6), DistanceEvent(7), ManualEvent(8)

#### 4.1.2.1 Format syntax

**Ex mTime <method> <eventNo> <newline>**

**Where**:

**x** is one of:

CycleBegin=1, StartOfLine =2,EnfOfLine =3,  EndOfSegment=4, Timeevent =5,

ExternalEvent=6,  DistanceEvent=7,  ManualEvent =8,  CycleEnd=9

**mTime**
Specifies the computer time of cycle as YYYY:DDD:HH:MM:SS.SS

**method**
1: port controlled, 2: time controlled (only used when E1).

**EventNo**
Sequence number for event. Not always in use.

Examples (Time event, StartOfLine):

```
E5 1997:324:09:02:32.000  2  0

E2 1997:243:12:34:23.345  2  0
```

#### 4.1.3 C (Cycle record)

This record will be saved when a new navigation cycle begins. Currently only **Cycle begin** (C1) will be stored (and only in General logging format).

A **Cycle begin** is received each time a new navigation cycle starts.

##### 4.1.3.1 Format syntax

Cx  mTime <cTime> <newline>

**Where**:

**x**

1: CycleBegin

**mTime**

Specifies the computer time of cycle begin as YYYY:DDD:HH:MM:SS.SS

**cTime**

Specifies the cycle time of last cycle as SS.ss (dec. seconds)

Example 1 (CycleBegin ):

```
C1 1997:243:12:34:23.345  0.50
```

#### 4.1.4 P (Position record)

A Position record can be of different types: ordinary positions (**O**), fixed positions = User defined offsets (**X**), dynamic positions (**D**) e.g. ROV,  filtered positions (**F**)

Future Option: Secondary positions (**S**)

##### 4.1.4.1 Format syntax

**P type nm Time Easting Northing Height LAT LONG DOP KP DAL DOL MC<newline>**

**Where**:

**type**

is 0, X, D, F or S - see above.

**nm**

if type= O: First nm holds reference position group number (1..5). If type = D (dynamic position) nm is the ‘offset’ number (= index in instrument array).

**Time**

Specifies the time of record as YYYY:DDD:HH:MM:SS.SS.. [float].

**Easting**

Easting in metres [float].

**Northing**
Northing in metres [float].
**Height**
The height in metres [float].
**LAT**
Option: custom format: Latitude of position (DDD.mmmmm).
**LONG**
Option: custom format: Longitude of position (DDD.mmmmm)
**DOP**
Dilution of position (The standard deviation).
**KP**
Kilometre point – 0 if no runline for position object is selected.
**DAL**
Distance Along Line – 0 if no runline for position object is selected.
**DOL**
Distance Of Line – 0 if no runline for position object is selected.
**MC**
Meridian Convergence in decimal degree (D.dddd)

##### 4.1.4.1.1 Example 1 (Primary Position):

```
P  O   1  2004:04:13:12:26:00.903   509999.5124  6299999.8980   0.4104  56.84370332
9.16392030   2.67   0.488  488.44  183.66  0.1372
```

##### 4.1.4.1.2 Example 2 (Dynamic position at ‘offset’ 3):

```
P  D   3  2004:04:13:12:25:59.325   583570.7451  6595765.3960  990.0000  59.49199504
10.47579652  25.39  302.543  302542.50  -40477.14  1.2715
```

##### 4.1.4.1.3 Example 3 (Fixed position at ‘offset’ 9 - eg position of winch):

```
P  X   9  2004:04:13:12:26:00.903   509999.5124  6299999.8980   0.0000  56.84370332
9.16392030   2.67   0.488  488.44  183.66  0.1372
```

### 4.1.5 A (Attitude record: RPH)

Contains Roll, Pitch and Heave at Time.

**Note**: Sensor can be placed on Object (eg a ROV).

#### 4.1.5.1 Format syntax

**A no index Time Roll  Pitch Heave <new line>**

**Where**:

**no**

0 if vessel sensor otherwise object number the sensor belongs to [integer].

**index**

Is sensor index (offset) for sensor. Sensor offsets are part of header record.
**Note**: This number is the index to the instrument array [integer].

**Time**

Specifies the time of record as YYYY:DDD:HH:MM:SS.SS. [float].

**Roll**

Specifies the vessel roll value used in LOP calculations of record in degrees [float]. Range: [-180 .. +180], default 0. Positive, when starboard sinks from horizontal plane.

**Pitch**

Specifies the vessel pitch value of record in degrees [float]. Range: [-180 .. +180] default 0. Positive, when bow raises from horizontal plane.

**Heave**

Specifies the vessel heave value of record in metres [float]. Default 0 metres. Positive, when ship is higher than horizontal plane.

**Example**:

```
A 0 15 1997:243:12:34:23.345 4.34 3.24 0.34
```

#### 4.1.6 D (Data acquisition record)

This record holds data acquisition data such as single-beam echo sounders, pipe tracker or plankton counters.

**Note**: Depths are not roll-, pitch-, heave-corrected.

##### 4.1.6.1 Format syntax

**D no index channel_number Time BathyDepth NumOfData Data$_1$ TimeAge$_1$ ..<new line>**

**No**

0 if vessel sensor, otherwise object number the sensor belongs to [integer].

**Index**

Sensor index (offset) for sensor. Sensor offsets are part of header record.
**Note**: This number is the index to the instrument array [integer].

**channel_number**

Is the data instrument channel number.

**Time**

Specifies the time of record as YYYY:DDD:HH:MM:SS.SS.. [float].

**BathyDepth**

Specifies the ROV bathy value of record in metres [float]. Default 0: metres.

**NumOfData**

Number of data records. Defined if >= 0.

**Data$_N$**

data N of record in given unit [float], default 0.

**TimeAge$_N$**

Time age from **Time** of depth N in milliseconds [float].

**Example** (depths from sensor offset 12):

```
D 0 12 1997:243:12:34:23.345 12.33 3 13.33 100 14.56 200 12.59 300
```

#### 4.1.7 G (Gyro record)

This record holds Gyro and CMG at Time.

**Note**: Sensor can be placed on Object (eg a ROV).

##### 4.1.7.1 Format syntax

**G no index  type Time Gyro CMG <new line>**

**Where**:

**no**

0 if vessel sensor otherwise object number the sensor belongs to [integer].

**index**

is sensor index (offset) for sensor. Sensor offsets are part of header record. NB: This number is the index to the instrument array [integer].

**Type**

Id number that identify the gyro

**Time**

Specifies the time of record as YYYY:DDD:HH:MM:SS.SS.. [float].

**Gyro**

Specifies the vessel gyro value of record in degrees [float]. Range: [0 .. 360], default 0. Positive clockwise.

**CMG**

Course Made Good. Id. the calculated heading for primary positions.  [float]. Range: [0 .. 360], default 0. Positive clockwise.

**Example**:

```
G 7 13 1997:243:12:34:23.345 231.34 230.71
```

**Note**: CMG is not saved in custom logging mode.

#### 4.1.8 R (Raw data record)

These records hold raw data strings from all instruments. The first item specifies the instrument type.

#### 4.1.8.1 Format syntax

**R TYPE index Time Data EOFRECORD<new line>**

**Where**:

**TYPE**
The raw instrument type. See instrument list below for details.

**Time**
Specifies the time of record as DDD:HH:MM:SS.SS.. [float].

**Data**
String that holds the raw data from the sensor.

**EOFRECORD**
Holds the string: **/R TYPE**

**Index**
Is sensor index (offset) for sensor. Sensor offsets are part of header record.
**Note**: This number is the index to the instrument array [integer

The following instruments are supported in NaviPac (4) online help from NaviPac Configuration.

```
/*========================================================================
                    SURFACE NAVIGATION  1 - 99
========================================================================*/
#defineTRACKPOINT_II_SUR          1
#define HPR300_SURF               2
#define HPR300P_SURF              3
#define HPR410P_SURF              4
#define SYLEDIS                   5
#define SYLEDIS_XYLL              6
#define ARTEMIS_RB                7
#define ARTEMIS_MK4               8
#define FANBEAM                   9
#define LBL_APS3_SURF             10
#define MOTOROLA_FALCON           11
#define LBL_SIMRAD_SURF           12
#define POLARTRACK                13
#define NAUTRONIX_SURF            14
#define ARTEMIS_RB_II             15
#define FANBEAM_II                16
#define MICROFIX                  17
#define MICROFIX_II               18
#define FREE_POSITION_INPUT       19
#define REMOTE_NP_SURF            20
#define SYLEDIS_SB5               21
#define LASER_METER               22
#define TRIMBLE_4000_1            24
```

```
#define TRIMBLE_4000_2          25
#define XY_POSITION             26
#define HYPERFIX                27
#define SEADIF                  28
#define CNAV1                   29
#define CNAV2                   30
#define TOPCON1                 31
#define TOPCON2                 32
#define TOPCON3                 33
#define TOPCON4                 34
#define POSMV_POS               35
#define SERCEL_NS230            36
#define NASNETR_POS             37
#define AGA_GEO                 38
#define ATS_3D                  39
#define LEICA_TOT1              40
#define LEICA_TOT2              41
#define ASHTECH_1               42   // Ashtech 1
#define ASHTECH_GPS             43      // Ashtech 2
#define GPS_NO1                 44      // NMEA GPS 1
#define GPS_NO2                 45      // NMEA GPS 2
#define LEICA1                  46      // Leica1: binary format
#define LEICA2                  47      // Leica2: binary format
#define GECO_GPS1               48
#define GECO_GPS2               49
#define UKOOA_GPS1              50
#define GECO_GPS3               51
#define GECO_GPS4               52
#define GECO_GPS5               53
#define TOPCONTOTAL1            54
#define TOPCONTOTAL2            55
#define GPS_NO3                 56      // NMEA GPS 3
#define GPS_NO4                 57      // NMEA GPS 4
#define GPS_NO5                 58      // NMEA GPS 5
#define Edgetech4600POS         59
/*=========================================================================
             GYRO   100 - 199
 =========================================================================*/
#define GYROHPR300              100
#define GYROHPR300P             101
#define GYROHPR400              102
#define DIGILOG_GYRO            103
#define NMEA1_GYRO              104
#define NMEA2_GYRO              105
#define NMEA3_GYRO              106
#define GYRONAUTRONIX           107
#define GYROSCSTRENCHER         108
#define GYROSCSROV              109
#define GYROSOLO                110
#define KVH_GYRO_TRACK          111
#define GYRO_LBL_SIMRAD         112
#define GYRO_TRITON_XL          113
#define GYRO_ROB_RSG4           114
```

```
#define GYROKONGSBERGSDP          115
#define GYRORDIPD0                116
#define GYROOCTANS                117
#define GYROPOSIDONIA             118
#define APS3_HEADING              119
#define NPR_HEADING               120
#define ANSCHUTZ_STD20            121
#define MANUAL_GYRO               122
#define INNOVATOR3_GYRO           123
#define BALDER_GYRO               124
#define TOPCON_GYRO               125
#define POSMV_GYRO                126
#define MDL_GYRO                  127
#define MINERLG_GYRO              128
#define GYRO_PSIMSSB              129
#define NASNETR_GYRO              130
#define O_PLOUGH_GYRO             131
#define PRDID_GYRO                132
#define HMR3000_GYRO              133
#define EM3000_GYRO_HQ            134
#define CDL_TOKIMEC2_G            135
#define Edgetech4600GYRO          136
#define PHINSGYRO                 138
#define SFIIIGYRO                 139
#define LOADSTAR_GYRO             140
#define ANSCHUTZ_NMEA             153
#define LEMKUHL_LR_40             154
#define GYRO_FROM_RTK             155
#define ROBERTSON_SKR80           156
#define SG_BROWN                  157
#define SEAPATH                   158
#define GYRO_FROM_2OBJ            159
#define LDTRAVOCEAN_GYRO          168
#define FREEGYRO1                 197
#define FREEGYRO2                 198
#define FREEGYRO3                 199

/*=======================================================================
              MOTION SENSOR 200 - 299
 ========================================================================*/
#define SEAPATH_MOTION            200
#define MOTION_HPR300             201
#define MOTION_HPR400             202
#define SEATEX_MRU                203
#define DIGILOG_MOTION            204
#define MOTIONNAUTRONIX           205
#define MOTIONSCSTRENCHER         206
#define MOTIONSCSROV              207
#define MOTIONSOLO                208
#define HIPPY_RPH                 209
#define DYNABASE_CRU_ATLAS        210
#define RPKVH_GYRO_TRACK          211
#define MOTION_LBL_SIMRAD         212
```

```
#define MOTION_TRITON_XL            213
#define MOTIONKONGSBERGSDP          215
#define MOTIONHYDRINS               216
#define MOTIONOCTANS                217
#define MOTIONPOSIDONIA             218
#define APS3_MOTION                 219
#define NPR_MOTION                  220
#define MANUAL_MOTION               222
#define INNOVATOR3_MOTION           223
#define BALDER_MOTION               224
#define TOPCON_RP                   225
#define POSMV_RPH                   226
#define MDL_RP                      227
#define MINERLG_RP                  228
#define NASNETR_RP                  230
#define O_PLOUGH_RP                 231
#define PRDID_RP                    232
#define HMR3000_RP                  233
#define EM3000_RPH_HQ               234
#define CDL_TOKIMEC2_RP             235
#define Edgetech4600RPH             236
#define TriaxusRP                   237
#define PHINSRPH                    238
#define SFIIIRP                     239
#define LOADSTAR_RP                 240
#define LDTRAVOCEAN_RP              268
#define TSS332                      276
#define TSSDMS05                    278
#define FREEMOTION1                 297
#define FREEMOTION2                 298
#define FREEMOTION3                 299

/*========================================================================
              SPEED MEASUREMENT 300 - 399
 ========================================================================*/
#define RDI_WORKHORSE_NAVIGATOR_DVL 300
#define CONSILIUM_SAL               362
/*========================================================================
              SUB-SEA AND REMOTE POSITIONING 400 - 499
 ========================================================================*/
#define LBL_APS3                    400
#define SGU_CABLELENGTH             401
#define LBL_SIMRAD                  402
#define WILD_1                      403
#define WILD_2                      404
#define SEAKING_RB                  405
#define DIGICOURSE_POS              406
#define REMOTE_FANBEAM              407
#define GOLF_LASER                  408
#define AGA_VESSEL                  409         // AGA from vessel
#define AGA_POINT                   410         // AGA from point
#define LEICA_DISTO                 411         // Distance measurement
```

```
#define HUGIN_AUV                   412
#define CABLELENGTH_2               413    // Identical to SGU_CABLELENGTH
#define REMOTE_FANBEAM2             414
#define SPERRE_BOX_POS              415
#define NASNETR_REMPOS              416
#define O_PLOUGH_REMPOS             417
#define SEATRACK_TAIL               418
#define AGA_POINT_II                419    // AGA from point - device ""
#define PIPEBUNDLE_POS              420    // EIVA Pipebundle position
#define REMOTE_NAV_1                421    // Remote dynamic object 1
#define REMOTE_NAV_2                422
#define REMOTE_NAV_3                423
#define REMOTE_NAV_4                424
#define REMOTE_NAV_5                425
#define REMOTE_NAV_6                426
#define REMOTE_NAV_7                427
#define REMOTE_NAV_8                428
#define REMOTE_NAV_9                429
#define REMOTE_NAV_10               430
#define FREEREMOTEPOS               431
#define FREEREMOTERELPOS            432
#define REMOTE_POS_3                433
#define REMOTE_POS_4                434
#define REMOTE_POS_5                435
#define REMOTE_POS_6                436
#define REMOTE_POS_7                437
#define REMOTE_POS_8                438
#define REMOTE_POS_9                439
#define REMOTE_POS_10               442
#define TRIMBLE_CB430               443
#define PHINSPOS                    444
#define RGPS_2                      445           // Like 418
#define FORSSEANAVCAM               446
#define REMTOPCONTOTAL1             454
#define REMTOPCONTOTAL2             455
#define LOADSTAR_POS                456
#define NAUTRONIXRS9_2              482
#define HPR300_2                    484
#define HPR410P_2                   486  // HiPap
#define iUSBL                       487  // HiPap alike telegram
#define ORE_BATS                    488  // ORE BATS
#define SONARDYNE_USBL              489  // $PSONUSBL
#define TRACKPOINT_II               491
#define NAUTRONIXRS9                492
#define POSIDONIA                   493
#define HPR300                      494
#define HPR300P                     495
#define HPR410P                     496  // HiPap
#define SIMRAD_HAIN                 497  // HiPap final calculated
#define REMOTE_POS_1                498
#define REMOTE_POS_2                499
```

```
/*=========================================================================
                    SPECIAL DATA INPUT 500 - 599
 =========================================================================*/
#define ZDA_TIME                        500
#define OSPREY_DP_WAYPOINT              501
#define SCS_ROV                         502
#define NMEA_TARGETS                    503
#define NMEA_WAYPOINT                   504
#define TRIMBLE_UTC_TIME                505
#define RAYTHEON_RADAR                  506
#define CONTROL_FROM_REMOTE_NP          507
#define LEM30                           508
#define STEINCO_ANCHOR_CAT              509
#define GENERIC_ANCHOR_CAT              510
#define RESON_MINEWARN                  512
#define SIMRAD_RDN50                    513
#define AIS_TARGETS                     514
#define SERCEL_NAUTILUS                 515
/*=========================================================================
                    DATA_ACQ (echo sounders): 700 - 799
 =========================================================================*/
#define ATLAS_DESO_25                   700
#define SIMRAD_EA300P                   701
#define NAVITRONIC_NAVISOUND_2000       702    //NaviSound 2000 (old NP:211-213)
#define NMEA_183_DEPTH                  703    //(old NP:247)
#define DIGIQUARTZ_BATHY                704
#define SIMRAD_BATHY                    705
#define ULVERTECH_BATHY                 706
#define ELAC_LAZ_4700                   707
#define MAG_G_880                       708    // 3 channels
#define ODOM_ECHOTRAC                   709 // 2 channels
#define FREE_IN_1                       710
#define FREE_IN_2                       711
#define FREE_IN_3                       712
#define SIMRAD_EA500                    713
#define SCSROV_DEPTH                    714
#define SCSTRENCHER_DATA                715
#define SOLO_DATA                       716
#define SIMRAD_EA200                    717
#define TRITECH_SCU3_BATHY              718
#define SUBSPECTION_CP                  719
#define CORROCEAN_CP                    720
#define MESOTECH_SCANNER                721
#define GSSI_GEORADAR                   722 // 1 channel
#define MAG_G_880_PRC                   723    // 3 channels
#define AANDERAA_PRES                   724 // 1 channel
#define DEPTH_TRITON_XL                 725 // 1 channel & 1 bathy
#define MAG_G_866                       726 // Raw, processed, normalised
#define GC_ROTS                         727 // 3 channels with ROTS information
#define RDI_CURRENT                     728 // 3 channels current X/Y and dir
#define BENTHOS_PSA_900                 729 // 3 channels depth / temp / range
```

```
#define DGH2611                    730 // 1 channel - speed
#define OCTANSTURN                 731 // 1 channel - no. turns
#define APS3_DAQ                   732    // 2 channels - altitude & salinity
#define FEATHER_ANGLE              733 // 11-3 channels
#define NPR_DAQ                    734 // 1 channel
#define DAQ_JETKNIFE               735 // 3 user defined channels
#define DAQ_ROTSII                 736    // 3 user defined channels
#define ELAZLAZ4721                737 // 1 channel per port
#define ELACHS4300                 738    // Up to three channels
#define NMEA_SPD                   739 // 2 channels CMG (deg) and SMG (knot)
#define DATAFROMSIS1000            740 // 3 channels (Depth, MAGx,MAGY, MAG)
#define DATAFROMSIS1000_II         741 // 3 channels (SV, Cond, Temp, {Range})
#define DATAFROMADCONV             742 // Up to 3 channels (From EIVA A/D)
#define DATAFROMSTD20              743 // 3 channels from ANschutz std. 20
(gyro, ROT, mag course)
#define CAPJET_IN1                 744 // 3 channels - SW Left, SW Right, hor.
dist
#define CAPJET_IN2                 745 // 3 channels - Speed1, speed2, Gyro
#define DEPTH_ADCP                 746 // 3 channels - Dpt1, Dpt2, Dpt3
#define INNOVATOR3_DAQ             747 // 2 Channels Alt + bathy
#define BALDER_DAQ                 748 // 2 Channels Alt + bathy
#define RESON_DESO                 749 // 3 Channels D1, D2, SV (and two
draft's)
#define TSS_340350                 750 // 3 channels dX, dZ, depth of cover
(Alt)
#define TSS_340350SS               751 // 3 channels avgSS,SSP,SSS
#define AR_PLOUGH1                 752 // 2 Channels Skid
#define AR_PLOUGH2                 753 // 3 Channels Depth
#define AR_PLOUGH3                 754 // 2 Channels Force/Tension
#define AR_CID1                    755 // 3 Channels Counter Port
#define AR_CID2                    756 // 3 Channels Counter Starboard
#define NMEA_MWV                   757 // 2 Channels Wind angle and speed
#define INNOVATUM_TRACKER          758 // 3 channels variable I/F
#define FJORDKABEL_DAQ             759 // 3 channels tension, length, speed
#define SCANFISH_DAQ               760 // 3 channels user selectable
#define SAIV204_A                  761 // 3 channels Pres, Temp, Cond
#define SAIV204_B                  762 // 3 channels Depth, Sal, Sv
#define DISTSHOOTINFO              763 // 3 channels distance, time to next,
time between
#define TRITECH_SCU3_BATHY2        764 // 3 channels salinity, temperature and
conductivity
#define STEINCO_ANCHOR             765 // 2 channels length and tension
#define NASNETR_DAQ                766 // 3 channels depth, pressure &
temperature
#define NMEA_MWV_P                 767 // 2 Channels Wind angle and speed
(Processed to absolute)
#define HIPAP_TRAVEL_TIME          768 // 2 Channels Travel time and range
#define O_PLOUGH_ASLAID            769 // 3 Channels Tow, DOB, Tension
#define O_PLOUGH_VAR               770 // 3 Channels Port Skid, Stb Skid,
Angle
#define O_PLOUGH_DATA              771 // 3 Channels depth, distance, steer
Angle
#define O_CAROUSEL                 772 // 3 Channels distance, speed. Tension
```

```
#define FREE_IN_4                 773
#define FREE_IN_5                 774
#define LDM41DAQ                  775 // 1 channel distance measurement
#define DILLON_DYNANOMETER        776 // 2 channels Dillon Dynamometers
EDxtreme
#define WETLABS_FLNTU             777 // 2 channels Fluorometer and Turbidity
#define DIGIQUARTZ_BAROMETER      778 // 2 Channels Barometer and Barometer
compensated
#define DIGIQUARTZ_BATHY_COMP     779 // Like Digiquartz - just compensated
for surface pressure (DIGIQUARTZ_BAROMETER)
#define VALEPORT_DAQ              780 // Valeport DAQ  Bathy, SV Den
#define VALEPORT_CTD              781 // Valeport proc 2 Sal Den Pres
#define VALEPORT_RAW1             782 // Valeport Raw 1 Cond, Temp, SV
#define VALEPORT_RAW2             783 // Valeport Raw 2 Pres
#define TRITECH_ALT               784 // 1 channel Altitude
#define SEABIRD_PRES              785 // 3 channels temperature and pressure
and resulting bathy depth
#define ODIM_WINCH                786 // 2 channels Cable length and tension
#define iUSBLQC                   787 // 3 channels pitch, yaw, calc length
#define EILERSEN_TENSION          788 // up to 3 channels T1 - T3
#define SEAMASTER_TRAWL           789 // 2 channels length and tension
#define FAHRENTHOLZ_DAQ           790 // 2 channels 15 and 100 KgHz
#define LEG_PENETRATION           791 // depth and penetration for 4 legs
(SeaJack)
#define JD_ANGLE                  792 // 1 channel with angle 0 - 8192 pulser
absolute
#define JD_CATERPILLAR            793 // 3 channels with data for JD
#define NPS_HYDROC                794 // NaviPac Science - HydroC measure CO2
#define NPS_CTDN                  795 // NaviPac Science - Teledyne CTD
#define NPS_CTDN2                 796 // NaviPac Science - Teledyne PSS
#define NPS_CYCLOPS               797 // NaviPac Science - Turner Cyclops
Measure Chl-a
#define NPS_GALOXO                798 // NaviPac Science - Galvanic Oxygen
#define NPS_AMTPH                 799 // NaviPac Science - AMT Ph value
#define NPS_TRIAXUS               850 // NaviPac Science - MacArtney Triaxus
#define NPS_TURNERC6              851 // NaviPac Science - MacArtney Turner
C6 Cyclops 7
#define NPS_AANDRAA4330           852 // NaviPac Science - MacArtney Aanderaa
4330 Optode
#define NPS_VALEPORT_ROVLOG       853 // Valeport ROVLOG Raw (Sx,Sy,Dir)
#define NPS_VALEPORT_ROVLOG_PROC  854 // Valeport ROVLOG Processed
(Speed,Dir)
#define DAQX_FREE                 855 // Free expanded DAQ
#define PS_30                     856 // Pressure/depth   (JD)
#define MACARTNEY_MINICOM40       857 //   (JD)
#define FREE_IN_6                 858
#define FREE_IN_7                 859
#define FREE_IN_8                 860
#define FREE_IN_9                 861
#define FREE_IN_10                862
#define SFIIIDAQ                  863    // Data from ScanFish III
#define VAISALA_WEATHER           864
#define MAG_G_882_MULTI           865    // 3 channels MAG, DEPTH, ALT
```

```
#define EMGS_WINCH                 866    // 3 channels Len Tension, Speed
#define EMGS_WINCH2                867    // 3 channels Angle FwdAft, Angle PortSTb
#define LDTRAVOCEAN_DEPTH          868 // LD TravOcean trencher depth
#define LDTRAVOCEAN_AUX            869 // LD TravOcean trencher DAQ/Aux data
#define FIXED_DEPTH                870    // fixed depth - manually entered
```

To get an up-to-date listing of instrument types, please request newest Instype.h from EIVA (support@eiva.com)

#### 4.1.8.2 Examples

##### 4.1.8.2.1 GPS (43) raw data:

Data format syntax:

ASCII NMEA 0183 GGA format (Global Positioning System)

$GPGGA,hhmmss.ss,xxxx.xxxxx,N,yyyyy.yyyyy,W,Q,ss,hh.h,uaaaaa,M,ugggg.g M,AAA,IIII

##### 4.1.8.2.2 Motorola (11) raw data:

NaviPac uses the Range Only Remote Range Record format:

<STX>HH,MM,SS.S,CC,RRRRRR.R,SS,CC,RRRRRR.R,SS,CC,RRRRRR.R,SS,CC,RRRRRRR,SS,TT,L,EEEE<ETX>

##### 4.1.8.2.3 Examples of raw data:

R42 1 1998:043:09:12:04.524
$GPGGA,081204.00,5609.43767,N,1013.50808,E,3,7,3.05,+00037.4,M,+00012.9,M,0.87,1021 /R42
R153 4 1998:043:09:12:04.535 $HEHDT,000.0,T/R153
R154 5 1998:043:09:12:04.536 $HEHDT,000.0,T/R154
R362 6 1998:043:09:12:04.538 $VDDPT,012.3,00.0/R362
R702 19 1998:043:09:12:04.569 *..900358400035739000583/R702
R700 20 1998:043:09:12:04.570 DA 66.28/R700

### 4.1.9 V (velocity record)

This record holds Speed/SMG information at Time.

**Note**: Sensor can be placed on Object (eg a ROV).

#### 4.1.9.1 Format syntax

**V no index Time V Vx Vy smg<new line>**

**Where**:

**no**;

0 if vessel sensor otherwise object number the sensor belongs to [integer].

**index**

is sensor index (offset) for sensor. Sensor offsets are part of header record.

**Note**: This number is the index to the instrument array [integer].

**Time**

Specifies the time of record as YYYY:DDD:HH:MM:SS.SS..

**V**

Specifies the vessel speed value of record in m/s. [float]. Range: [0 .. 99], default 0 if no speed log.

**Vx**

Velocity vectors X component in ±m/s [float], default 0.

**Vy**

Velocity vectors Y component in ±m/s [float], default 0.

**SMG**

Speed Made Good. Id. the calculated speed for primary positions in m/s [float].

Example:

```
V 3 8 1997:243:12:34:23.345 2.00 3.00 3.606 3.709
```

#### 4.1.10 S (GPS Status record)

The S record gives (if available) expanded UKOOA information for attached GPS receivers:

##### 4.1.10.1 Format syntax

**S no Time noSat HDOP antH VDOP PDOP err95maj err95min err95ori err95east err95north err95heigh extRel unitvar agecorr noRef quality (fixstate) lat long prn<new line>**

**Where**:

no : Instrument index number. See page 43 for detailed list.
Time: Timestamp of GPS status
NoSat: Number of satellites
HDOP: Horizontal dilution of precision
AntH: Antenna height;
VDOP: Vertical dilution of precision
PDOP: PDOP
err95maj: UKOOA - Error ellipse 95 % semimajor
err95min; UKOOA - Error ellipse 95 % semiminor
err95ori; UKOOA - Error ellipse 95 % orientation
err95east; UKOOA - Error ellipse 95 % easting
err95north; UKOOA - Error ellipse 95 % northing
err95heigh; UKOOA - Error ellipse 95 % height

extRel; UKOOA -  External reliability
unitvar; UKOOA -  Unit variance
agecorr: Age of correction
noRef: Number of reference stations
quality: Ordinary GPS(1) Differential (2) or RTK (3)
fixstatus: Raw fix status from telegram (eg. NMEA GSA telegram or Geco/UKOOA special telegram)
lat: Antenna position latitude. Given as decimal degree ddd.ddddddddd including sign. Always in WGS84.
long: Antenna position longitude. Given as decimal degree ddd.ddddddddd including sign. Always in WGS84.
Prn: Bit pattern defining list of used satellites. Bit 0 represents satellite 1 etc.

Parameters labelled UKOOA are either calculated or read using standard UKOOA definitions.

**Example**:

S 48 2005:01:21:08:25:07.907 7 1.20 100.10 2.10 2.42 5.99 0.54 89.99 0.43 4.80 1.40 0.54 0.16 0.00 3 2 (3) 56.646567833 3.875218833 21496897

## 4.2 Survey format (NaviEdit)

The purpose of the survey-logging format is to be able to make charts using the NaviPac interpreter in NaviEdit. From here, data can be exported to charting software (eg as XYZ).

The reason having a separate file for NaviEdit interpretation is to avoid large amount of data with no relevance for this purpose (eg raw instrument data).

It is recommended, however, to enable the general logging format until you are sure everything is set up correctly. This could be used to validate different offsets, events, and raw data if anything goes awry.

The Survey format is contained in a plain ASCII file, therefore in Windows each line is terminated by <CR><LF>.

The logfile consists of some general header records with some common logging information (eg logging version number, user info, projection, datum shift, sensor offsets) followed by all Navigation records (eg reference position, relative offset positions (eg position off ROV with echo sounder or position of echo sounder on vessel), attitudes, SMG, CMG).

**All** items are always separated by a separator **semicolon** '**;**'

The data format is structured in records of data items. These records are described below.

The first character of each record defines the type of record:

H (Header record),
P (Position record),
D (Data acquisition record) (eg Depths or magnetometer)
A (Attitude record: RPH)
G (Gyro record)
V (Velocity)
S (GPS status info)

All records will include date and time formatted like:

YYYY:MM:DD::HH:MM:SS.SS

**Note**: X,Y,Z files should be logged in custom format.

### 4.2.1 Survey format (NaviEdit) record types

These headers hold the following information:

Start of header ID =H0 – H5
Custom logging format version ID (H0)
User info (H0)
Log file name (H0)
Time, date, year of log file creation (H0)
Projection parameters (H1)
Ellipsoid parameters (H2)
Datum shift parameters (H3)
Sensor offsets (H4x) – x is channel number 1-3 for echosounders
Sensor CO values (H5)
User defined offsets (H4)

These headers will be generated by the logging program (by reading the current setup) each time logging is started or a new runline is selected.

#### 4.2.1.1 Instrument types

In headers and data records an instrument type will be logged:

SURFACE NAVIGATION 0 - 99
GYRO 100-199
MOTION SENSORS 200-299
SPEED MEASUREMENT 300 – 399
SUB-SEA AND REMOTE POSITIONING 400-499
DATA INPUT 500 – 599 and 900-950
DATA OUTPUT 600 – 699
DATA_ACQ (echo sounders/User def. inputs) 700 – 799

OFFSETS: 800 - 860

#### 4.2.2 H0 (primary header)

This is the primary header for the NaviEdit Survey recording format. It holds information about the current logfile and the syntax used.

##### 4.2.2.1 Format syntax

**H0; Time; logversion; logfile; user info/H0<new line>**

**Where**:

**Time:** Specifies the time of record as YYYY:MM:DD:HH:MM:SS
**Logversion:** version string: log file version
**Logfile:** The name of the logfile (incl. Path)
**user info:** NaviEdit(NT) Interpreter Format
**/H0:** End of primary header

Additional header information (H0) holds a description of the record formats (P, D, A, G)

**Example**:

H0; 2000:10:05:11:05:49.000; 3.2.0; C:\Eiva\NaviPac\log\001005\001005S002.npd; NaviEdit(NT) Interpreter Format /H0

H0; P; O; no; Time; E; N; height; stddev; ; ; ; kp; dal; dol; fix /H0

H0; P; D; no; Time; E; N; height; stddev; dx; dy; dz; kp; dal; dol /H0

H0; D; no; insttype; index; channel_number; Time; BathyDepth; NumOfDepths; Depth1; TimeAge1../H0

H0; A; no; index; Time; Roll; Pitch; Heave /H0

H0; G; no; index; type; Time; Gyro; CMG /H0

#### 4.2.3 H1 (projection parameters)

This is an additional header for the NaviEdit Survey recording format. An interpreter can read the current projection parameters here.

##### 4.2.3.1 Format syntax

**H1; projectionname; type; OrgScale; 1.Parallel; 2.Parallel; Longitude; Latitude; Easting; Northing; UTMzone; /H1<new line>**

Where **Longitude** and **Latitude** are defined in radians.

**Example**:

```
H1; UTM (north);  5; 0.999600000000; 0.575958653158; 0.785398163397;
0.157079632679; 0.000000000000; 500000.000000000000; 0.000000000000; 32 /H1
```

### 4.2.4 H11 (Position data unit)

This header identifies the data unit used for position (easting and northing)

#### 4.2.4.1 Format syntax

**H11; POS; Scaled; Name; Scale; unit /H11<new line>**

Where **Scaled** identifies if scaling is used or not. Scale determines the factor to convert to metric.

**Example**:

```
H11; POS; 1; Metric; 1.0000000000; m /H11
```

### 4.2.5 H12 (Depth data unit)

This header identifies the data unit used for data acquisition (depth, height, heave etc).

#### 4.2.5.1 Format syntax

**H11; DEP; Scaled; Name; Scale; unit /H11<new line>**

Where **Scaled** identifies if scaling is used or not. Scale determines the factor to convert to metric.

**Example**:

```
H12; DEP; 1; Metric; 1.0000000000; m /H12
```

### 4.2.6 H2 (Ellipsoid parameters)

This is an additional header for the NaviEdit Survey recording format. An interpreter can read the current ellipsoid parameters here.

#### 4.2.6.1 Format syntax

**H2; Ellipsoid; Inv.Flat; SemiMajorAxis /H2 <new line>**

**Example**:

```
H2; ED 50; 297.0000; 6378388.0000 /H2
```

### 4.2.7 H3 (Datum shift parameters)

This is an additional header for the NaviEdit Survey recording format. An interpreter could read the current datum shift parameters here.

#### 4.2.7.1 Format syntax

**H3; Tx; Ty; Tz; Rx; Ry; Rz; PPM; Method, Name /H3<new line>**

Transformation is given in metres and rotation in degrees.

NaviPac supports the following shift methods:

- 0 No Shift
- 1 Normal 7-parameters (Bursa Wolf)
- 2 Special North Sea method (2 step)
- 3 Modified Bursa Wolf
- 4 US NADCON

**Example**:

```
H3;    89.5000;    93.8000;    123.1000;    0.000000000000;    0.000000000000;
0.000043333333; -1.200000; 1 ; WGS84 to ED50 (Denmark)/H3
```

### 4.2.8 H31 (Local time reference system)

Defines how the recording computer was set up time-wise-

**Example**:

```
H31; Romance Standard Time; Romance Daylight Time;Zone;-3600;DL;1 /H3
```

### 4.2.9 H4 (Sensor offsets)

This is an additional header for the NaviEdit Survey recording format. An interpreter is able to read the current sensor offsets parameters (instrument type, index, object, X,Y,Z) here. Note the index to the instrument in the general set-up database (gensetup.DB).

#### 4.2.9.1 Format syntax

**H4<ch>; InstrumentName/objectName; instrument_type; index; objectNumber; Xoffset; Yoffset; Zoffset; [Transponder code] [;DAQ Type][;Assigned] /H4<ch> <new line>**

If the instrument is a dynamic positioning via a USBL system, the objects positioned by this instrument can have their own offsets and extra headers with the channel number <ch> inserted.

**Where**:

**Ch:** (optional) is channel number for echo sounder channels
**InstrumentName/objectName:** Is the name of the instrument/object
**instrument_type:** Is the instrument type for sensor
**index:** Is the index to the instrument array [integer]
**objectNumber:** Is the object number the instrument belongs to (e.g main vessel no 0 or ROV no 21).
**Xoffset , Yoffset, Zoffset:** offset values for instrument.
**[Transponder code]:** To be able to identify which transponder or calculation was used

- > 0 Transponder code for eg USBL, LBL
- -1 Combined position based on more objects
- -2 Final position from single source like remote GPS

**[DAQ Type]:** Identification of data acquisition type

- 0 Single beam echosounder / altimeter
- 1 Bathy depth
- 2 Magnetometer
- 3 Other
- 4 Pressure

**[Assigned]**

Lists up to 5 objects the unit (gyro/motion sensor) is used for. No 1 is identical to objectNumber. -1 defines not used.
**Note**: the 5 items are comma-separated.

**Examples**:

```
H4; GPS1 (NMEA); 44;  0;    0; 0.000; 0.000; 0.000; 0; -1,-1,-1,-1,-1 /H4
H4; NMEA1 Gyro; 104;  1;    0; 0.000; 0.000; 0.000; 0; 0,-1,-1,-1,-1 /H4
H4; NMEA1 Gyro; 104;  2;   20; 0.000; 0.000; 0.000; 0; 20,-1,-1,-1,-1 /H4
H4; TSS DMS05/TSS 335; 278;  4;    0; 0.000; 0.000; 0.000; 0; 0,-1,-1,-1,-1 /H4
H4; Atlas Deso 20/25; 700;  9;    0; 0.000; 0.000; 0.000; 0; -1,-1,-1,-1,-1 /H4
H41; A1; 700;  9; 7000; 0.120; 0.000; -4.300; 0 /H41
H42; A2; 700;  9; 7001; 0.120; 0.000; -4.300; 0 /H42
```

**Example** with USBL system (Transponders 1 used) and SPRINT (remote GPS)

```
H4; Kongsberg HiPAP/APOS; 496;  5;    0; 0.000; 0.000; 0.000; 0; -1,-1,-1,-1,-1 /H4
```

H4; SPRINT; 498;  6;    0; 0.000; 0.000; 0.000; 0; -1,-1,-1,-1,-1 /H4
H4; ROV [TP1]; 496;  5; 14201; 0.000; 0.000; 0.000; 1 /H4
H4; ROV; 496;  5;   21; 0.000; 0.000; 0.000; -1 /H4
H4; Towfish [SPRINT]; 498;  6; 14498; 0.000; 0.000; 0.000; -2 /H4
H4; Towfish; 498;  6;   20; 0.000; 0.000; 0.000; -1 /H4

Please note that a position source will be represented as the raw observation object (a 5 digit internal number like 14201) and the user selectable object (eg 21 for the ROV) representing the final position based on one or more sources

#### 4.2.10 H4 (Pipe and Cable tracker)

When using a cable and pipe tracker (eg TSS 340) a second header will be added to the system, as this enables NaviEdit to recognise the data as pipe and cable and to get maximum utilisation.

Format will be the same as for ordinary data acquisition – but the contents will be specialised:

H4; TSS 340/350; 750; 21;   20; 0.000; 0.000; 0.000; 0; -1,-1,-1,-1,-1 /H4
H41; PdX; 750; 21; 7500; 0.000; 0.000; 0.000; 3 /H41
H42; PdZ; 750; 21; 7501; 0.000; 0.000; 0.000; 0 /H42
H43; pB; 750; 21; 7502; 0.000; 0.000; 0.000; 0 /H43
H4; TSS 340/350-RAW; 7750; 21;   20; 0.000; 0.000; 0.000; 0; -1,-1,-1,-1,-1 /H4
H41; TSS 340/350; 7750; 21;   20; 0.000; 0.000; 0.000; 5 /H41

The 4 upper lines is the ordinary data acquisition header – and the lower 2 is the special lines added for the pipe/cable. The only differences are:

- The instrument type will be defined in the range 7700 and up (ordinary ID plus 7000)
- The DAQ type is defined as 5 which defines raw pipe data

#### 4.2.11 H4 (User-defined offset / POI – point of interest)

This special H4 header identifies all user-defined offsets in use.

**H4; objectName; instrument_type; index; BaseObjectNumber; Xoffset; Yoffset; Zoffset; Offset type  /H4<new line>**

**Where**:

**Object Name**
The name defined by operator

**Instrument_type**
Instrument type (801 to 840)

**Index**

Index in general setup database

**BaseObjectNumber**

Identifies which object the offset is define, e.g. 0 = vessel etc.

**Offset types**

Identifies the type of offset (ie how is the position calculated):

- MO_GYRO = 0
  Ordinary 3D calculation using vessel/object gyro
- MO_CMG = 1
  Ordinary 3D calculation using vessel/object course made good
- MO_DRAG = 2
  Dynamic user defined offset using drag (chain model)
- MO_FILTER=3
  Offset represent filtered position of ordinary object – e.g. filtered ROV position
- MO_SECONDARY=4
  Offset calculated as template tracking based on multiple sources – please see dedicated manual.
- MO_DEPTHDRAG=5
  Dynamic user defined offset using drag (chain model) compensated for water depth
- MO_CIRCULARAVG=6
  object towed behind two vessels

**Examples**:

H4; OffsetPåROV2; 801;  15;   4; 1.000; 2.000; 3.000; 0 /H4
H4; OffsetOnMyVessel; 802;  16;   0; 0.000; -12.000; 0.000; 1 /H4

### 4.2.12 H5 (Sensor C-O values)

This is an additional header for the NaviEdit Survey recording format. An interpreter is able to read the current instrument C-O parameters. There will be C-O values for motion, gyro and speedlog sensors. The index to instrument “array” in the general set-up database.

#### 4.2.12.1 Format syntax

Gyro/Motion:

**H5; InstrumentName; instrument_type; index; objectNumber ; no_of_CO_values; CO1; CO2; CO3; Latency /H5 <new line>**

**DVL**
**H5; InstrumentName; instrument_type; index; objectNumber; 1; 0.0; 0.0; 0.0; GYRO-MOUNT; ROLL-MOUNT; PITCH-MOUNT /H5**

**Where:**

**InstrumentName:**
Is the name of the instrument.

**instrument_type:**
Is the instrument type for sensor

**index**:
Is the index to the instrument array [integer].

**no_of_CO_values:**
Specify the number of C-O values following for sensor at index.

**CO1-C03:**
From 1-3 C-O values (E.g. Motion sensor () could have 3 CO (roll, pitch, heave)

**Latency:**
Data delay as defined in setup. Time stamp has been compensated

**GYRO-MOUNT; ROLL-MOUNT; PITCH-MOUNT:**
Mounting angles for DVL unit.

**Examples**:

```
H5, Anschutz NMEA 0183; 153;  1;    0; 1; 0.005; 0.000 /H5
H5; TSS 332; 276;  2;    0; 3; 0.000; 0.000; 0.000; 0.010 /H5
H5; RDI Workhorse Navigator DVL; 300;  4;    1; 1; 0.0; 0.0; 0.0; 1.000; 2.000;
3.000 /H5
```

### 4.2.13 H6 (Position system group)

This is an additional header for the Survey format (NaviEdit). A NaviPac interpreter can read the current navigation (position) groups (GroupNo, GroupName here).

#### 4.2.13.1 Format syntax

**H6; GroupNo; GroupName /H6<new line>**

If the instrument belongs to the group Dynamic Positioning via an USBL system, the objects positioned by this instrument can have their own offsets and extra headers in which <ch> will be inserted.

**Where**:

**GroupNo:** is the navigation group number (1-5).
**GroupName:** is the name of the navigation group – normally = first navigation system in the group.

**Examples**:

H6; 1; GPS1; /H6
H6; 2; Motorolla; /H6
H6; 3; GPSNmea; /H6

#### 4.2.14 Hx (Extra header group)

If NaviPac is operated together with a Digicourse tail buoy positioning system, then a dedicated **Hx** header is added.

##### 4.2.14.1 Format syntax

**Hx; InstrumentName; instrument_type; index; RefObjectNumber ; Off-X, Off-Y, Off-Z /Hx <new line>**
**Hx1; Bird; instrument_type; index;  no. birds, len1, len2,…,lenxx /Hx1<new line>**

**Where**:

**InstrumentName:**
Is the name of the instrument.

**instrument_type:**
Is the instrument type for sensor – always 406

**index**:
Is the index to the instrument array [integer]

**RefObjectNumber:**
Which object the Digicourse is attached to – always 14 -> Cable offset.

**Off-X, Off-Y, Off-Z:**
**3D o**ffset from vessel reference position to cable offset position.

**Bird:**
Always the text Bird.

**No. birds:**
Number of birds in use (1 – 16)

**Len1, .., lenxx:**
How far out is bird number I, given a length in metres from previous bird.

**Example**:

Hx; Digicourse; 406; 10;   14; 0.000; 0.000; 0.000 /Hx
Hx1; Bird; 406; 10;  6;50.00;50.00;50.00;50.00;50.00;50.00 /Hx1

#### 4.2.15 H7 (Special header)

To ease reading of survey files by external programs, we have added a summary object header H7, which lists the number of active objects in NaviPac:

##### 4.2.15.1 Format syntax

H7; Number of objects = **<Number of objects>** /H7

H7; **<Name>; <Object number>;<Instrument>; <Relative to>; <Offset X>; <Offset Y>;<Offset Z> ; <Offset type>** /H7 (One line per active object)

**Where**:

**Number of objects**

How many active objects are handled in NaviPac right now.

**Name**

Name of current object – as defined in set-up (objects.TXT)

**Object number**

Internal object number (ties names and numbers together in objects.txt)

**Instrument**

Which instrument is this object related to – see Instype.H for further details.
(eg. -1 –> No instrument and 496 -> Hipap UW positioning)

**Relative to**

For offsets – which basis object is this one based on.

**Offset X/Y/Z**

XYZ offsets.

**Offset type**

How is the offset/object calculated:
-1: No calculation based on basis object special handling (e.g. vessel position)
0: Offsets via gyro and motion data
1: Offset via CMG
2: Offset via drag method (chain emulation)
3: Filtered position
4: Combined position of two or more primary positions
99: Direct calculated (eg USBL objects)

**Example**:

H7; Number of objects = 8 /H7
H7; Vessel; 0; -1; 0; 0.000; 0.000; 0.000; -1 /H7
H7; USBL reference; 11; 496; 0; 0.000; 0.000; 0.000; 0 /H7
H7; Ref; 2; 496; 11; 0.000; 0.000; 0.000; 99 /H7
H7; Ref+10; 3; 496; 11; 0.000; 10.000; 0.000; 99 /H7
H7; SO-Tracks; 36; 421; 0; 0.000; 0.000; 0.000; 99 /H7
H7; NaviBat Depth; 7030; 703; 0; 0.000; 0.000; 0.000; 0/H7
H7; Ref+10-10; 801; 801; 3; 0.000; -10.000; 0.000; 0 /H7
H7; Offset på Vessel (Y+100); 802; 802; 0; 0.000; 50.000; 0.000; 0 /H7

### 4.2.16 P (Position record)

A survey format (NaviEdit) Position record can be of different types: ordinary positions (**O**)= reference position, dynamic positions (**D**) e.g. ROV on USBL system or Offset (**X**)) .

#### 4.2.16.1 Format syntax for Ordinary (O) and Dynamic (D) positions

**P; type; subtype; Time; Easting; Northing; Height; StdDev; dX; dY; dZ; KP; DAL; DOL; FIX; MC <newline>**

#### 4.2.16.2 Format syntax for Offsets (X positions)

**P; X; nm; Time; Easting; Northing; Height; LAT; LONG; StdDev; KP; DAL DOL; FIX; MC**

**Note**: Offsets are only logged if **Extra** check box in setup menu is selected.

**Where**:

**type:**
O, D, X - see above.

**subtype:**
if type= O: it holds reference position group number (1..5). If type is D (dynamic position) it holds the “offset” number (object number in header)

**Time:**
Specifies the time of record as YYYY:MM:DD:HH:MM:SS.SS.

**Easting:**
Easting in metres [float]. Set to ”” (empty) if type is D.

**Northing:**
Northing in metres [float]. Set to ”” (empty) if type is D.

**Height:**
The height in metres [float].

**StdDev:**
The standard deviation [float].

**dX:**
Relative X offset from reference position. Set to ”” (empty) if type is O or F.

**dY:**
Relative Y offset from reference position. Set to ”” (empty) if type is O or F.

**dZ:**
Relative Z offset from reference position. Set to ”” (empty) if type is O or F.

**KP:**
Kilometre point . Set to ”” (empty) if not computed (Helmsman).

**DAL:**
Distance Along Line. Set to ”” (empty) if not computed (Helmsman).

**DOL:**
Distance Of Line. Set to ”” (empty) if not computed (Helmsman).

**FIX:**
Event number from NP. Will repeat last event number if not updated.

**MC:**
Meridian Convergence in decimal degree (D.dddd)

**Example 1 (Primary Position - KP, DAL, DOL)**:

```
P; O;  1; 2004:04:13:12:25:57.903;  509999.5124; 6299999.8980;  0.4104;
2.60; ; ; ;   0.4884; 488.44; 183.66; 131; 0.1372
```

**Example 2 (Dynamic position at “offset” 3, dx, dy, dz)**:

```
P; D;  3; 2004:04:13:12:25:55.320;  583570.7451; 6595765.3960; 990.0000;
25.39;  0.0000;  0.0000;  0.0000; ; ; ; -9999; 1.2715
```

**Example 3 (offset position)**:

```
P; X;  9; 2004:04:13:12:25:56.902;  509999.5124; 6299999.8980;  0.0000;
56.8437033;  9.1639203;  2.58;  0.000;   0.00;   0.00; -9999; 0.1372
```

#### 4.2.17 A (Attitude record: RPH)

Contains Roll, Pitch and Heave at Time.

**Note**: Sensor can be placed on Object (eg a ROV).

##### 4.2.17.1 Format syntax

**A; no; index; Time; Roll; Pitch; Heave <new line>**

**Where**:

**no** :
0 if vessel sensor otherwise object number the sensor belongs to [integer].

**index**:
is sensor index (offset) for sensor. Sensor offsets are part of header record.
**Note**: This number is the index to the instrument array [integer].

**Time:**
Specifies the time of record as YYYY:MM:DD:HH:MM:SS.SS.

**Roll**:
Specifies the vessel roll value used in LOP calculations of record in degrees [float]. Range: [-180 .. +180], default 0. Positive, when starboard sinks from horizontal plane.

**Pitch**:
Specifies the vessel pitch value of record in degrees [float].Range: [-180 .. +180] default 0.Positive, when bow raises from horizontal plane.

**Heave:**
Specifies the vessel heave value of record in metres [float]. Default 0 metres. Positive, when ship is higher than horizontal plane.

**Example**:

```
A;    0;  4; 1999:03:19:15:19:17.072; -0.7200; -0.4500;  0.0000
```

#### 4.2.18 D (Data acquisition record)

This record can hold any data acquisition input such as single-beam echo sounder, magnetometer or plankton counters.

**Note**: Depths are not RPH corrected.

#### 4.2.18.1 Format syntax

**D; no; insttype; index; channel_number; Time; BathyDepth; NumOfValues; Value$_1$; TimeAge$_1$ ..<new line>**

**no** :
0 if vessel sensor otherwise object number the sensor belongs to [integer].

**Insttype:**
The instrument type. See header for instrument name.

**index**:
is sensor index (offset) for sensor. Sensor offsets are part of header record.
**Note**: This number is the index to the instrument array [integer].

**channel_number:**
Is the instrument channel number (1-3).

**Time:**
Specifies the time of record as YYYY:MM:DD:HH:MM:SS.SS

**BathyDepth**:
Specifies the ROV bathymetry value of record in metres [float]. Default 0 metres.

**NumOfValues:**
Number of data values in record [integer], default 0. Defined if >= 0.

**Value$_N$**:
Value N of record in defined data unit, default 0.

**TimeAge$_N$**:
Time age from **Time** of value N in milliseconds [float]. Default 0, defined if >= 0.

**Note on BathyDepth**: (used in 3 different way):

1. **ROV-mounted system with OSP system (like eg Tritech)**
   Here the BathyDepth will hold the measured bathymetry depth (ie the ROV's depth and the data part (DepthN) will hold the echo sounder depth).
2. **ROV-mounted system (without bathymetry sensor)**
   Here the BathyDepth will hold the measured depth (Z) from the USBL system (ie the best guess of the ROV's depth and the data part (DepthN) will hold the echo sounder depth).
3. **Vessel-mounted system**
   Here BathyDepth is equal to offset (corrected for RPH)

**Example** (depths from sensor mounted on vessel (no=0) with instrument index=8 and offset 702):

```
D;    0;  702;  8; 1; 1999:03:19:15:19:16.220;  0.5328; 2;35.8900;  0;35.4600;  29;
D;    0;  702;  8; 2; 1999:03:19:15:19:16.220; -2.4639; 2;35.6100;  0;35.3100;  29;
D;    0;  702;  8; 3; 1999:03:19:15:19:16.220; -5.4783; 2;-5.6100;  0;-5.7200;  29;
```

#### 4.2.19 G (Gyro record)

This record holds Gyro and CMG at Time.

**Note**: Sensor can be placed on an Object (eg a ROV).

##### 4.2.19.1 Format syntax

**G; no; index; Time; Gyro; CMG <new line>**

**Where**:

**no** :
0 if vessel sensor otherwise object number the sensor belongs to [integer].

**Index**:
Is sensor index (offset) for sensor. Sensor offsets are part of header record.
**Note**: This number is the index to the instrument array [integer].

**Time**: Specifies the time of record as YYYY:MM:DD:HH:MM:SS.SS.. [float].

**Gyro**: Specifies the vessel gyro value of record in degrees [float]. Range: [0 .. 360], default 0. Positive clockwise.

**CMG**: Course Made Good. Id. the calculated heading for primary positions. [float]. Range: [0 .. 360], default 0. Positive clockwise.

**Example**:

G; 0; 2; 1999:03:19:15:19:17.052; 200.0000; 198.8667

**Note**: CMG is not saved in custom logging mode.

#### 4.2.20 S (GPS Status record)

The S record gives (if available) expanded UKOOA information for attached GPS receivers:

##### 4.2.20.1 Format syntax

**S; no; Time; noSat; HDOP; antH; VDOP; PDOP; err95maj; err95min; err95ori; err95east; err95north; err95heigh; extRel; unitvar; agecorr; noRef; quality (fixstate); lat; long; prn<new line>**

**Where**:

**no** :
Instrument index number.

**Time**:
Timestamp of GPS status

**NoSat**:
Number of satellites

**HDOP**:
Horizontal dilution of precision
**AntH**:
Antenna height
**VDOP**:
Vertical dilution of precision
**PDOP**:
PDOP
**err95maj**:
UKOOA - Error ellipse 95 % semimajor
**err95min**: UKOOA - Error ellipse 95 % semiminor
**err95ori**: UKOOA - Error ellipse 95 % orientation
**err95east**: UKOOA - Error ellipse 95 % easting
**err95north**: UKOOA - Error ellipse 95 % northing
**err95heigh**: UKOOA - Error ellipse 95 % height
**extRel**; UKOOA - External reliability
**unitvar**; UKOOA - Unit variance
**agecorr**:
Age of correction
**noRef**:
Number of reference stations
**quality**:
Ordinary GPS(1) Differential (2) or RTK (3)
**fixstatus**:
Raw fix status from telegram (eg NMEA GSA telegram or Geco/UKOOA special telegram)
**lat**:
Antenna position latitude. Given as decimal degree ddd.ddddddddd including sign. Always in WGS84.
**long**:
Antenna position longitude. Given as decimal degree ddd.ddddddddd including sign. Always in WGS84.
**Prn**: Bit pattern defining list of used satellites. Bit 0 represents satellite 1 etc.

Parameters labelled UKOOA are either calculated or read using standard UKOOA definitions.

**Example**:

S; 42; 2001:01:24:16:30:56.000; 5; 3.04; 37.40; 0.99; 1.23; 32.10; 21.44; 98.75; 25.56; 17.42; 37.61; -1.00; -1.00; 0.87; -1; 3 (3); 23.12332145; 65.17000034; 794[2]

[2] PRN 794 corresponds to bit pattern 1100011010 (ie satellite 2,4,5,9,10)

#### 4.2.21 R (Raw Pipe and Cable tracker)

The R record gives (if available) the pipe and cable tracker (eg from TSS) as a raw data string, so the entire data can be loaded into NaviEdit for full utilization.

**Note**: This is only available if recorded using **NPRec**.

##### 4.2.21.1 Format syntax

**R; Obj no; Instrument Type; Index; Time; RAW <new line>**

| | |
|---|---|
| **Obj no** | The object number |
| **Instrument Type** | The instrument type number. See page 43 for detailed list. |
| **Index** | Instrument index – index in the list |
| **Time** | Time stamp |
| **RAW** | The raw instrument string |

**Example**:

R; 20; 750; 21; 2012:02:06:13:53:19.343;:S +0008 0135 0107+0028-001 3594 2394 3214 2554 6703 5573 00

#### 4.2.22 X (Digicourse streamer record)

For 2D seismic purpose, we have added a special data-set giving recorded data from digicourse.

##### 4.2.22.1 Format syntax

**X; DIGICOURSE; G; Index; time; no birds; from; to; compass data**

**X; DIGICOURSE; D; Index; time; no birds; from; to; depth data**

**Where**:

**G/D:**
indicates if it is compass (G) or depth data

**Index:**
is sensor index (offset) for sensor. Sensor offsets are part of header record.
**Note**: This number is the index to the instrument array [integer].

**Time:**
Time and date of data

**No birds**:
Number of birds in data-set (1-32)

**From:**
Identifies where the streamer starts. Given by object number (default 14)

**To:**

Identifies the endpoint (tail buoy) of the streamer. Given by object number (default 114)

**Compass data:**

Gives compass for each active bird – bird number one is closest to the vessel. 999.99 -> No compass.

**Depth data:**

Gives depth of each active bird – bird number one is closest to the vessel. 0.00 -> No depth available.

**Example**:

X; DIGICOURSE; G; 7; 2001:01:24:08:47:02.000; 19; 14; 114; 999.99; 999.99; 999.99; 999.99; 999.99; -5.50; -4.40; -3.30; -2.20; -1.10; -1.00; 999.99; 1.20; 2.30; 3.40; 4.50; 5.70; 6.70; 7.80;

X; DIGICOURSE; D; 7; 2001:01:24:08:47:02.000; 19; 14; 114; 0.00; 0.00; 0.00; 0.00; 12.34; 12.35; 0.00; 12.37; 12.38; 12.39; 12.40; 12.41; 12.42; 12.43; 12.44; 12.45; 12.46; 12.47; 12.48;

## 4.3 Custom format (Custom Logfile)

The custom NaviPac logging format will normally be used to log data that has relevance for a specific user. The custom format is designed for simpler use in, for example, Microsoft Excel, or NaviEdit via simple XYZ import.

The data format is contained in a plain ASCII file, therefore in DOS/Windows each line is terminated by <CR><LF>.

The log file can hold of a lot of different navigation information (eg positions, offset positions (like, for example, position on a ROV), attitudes, echo sounder depths). If the non-record-based format was selected, **LogData** simply waits to store a new line in the logfile until all the wanted/included data items arrive from the Kernel. As positions, depths and attitudes arrive asynchronised, these items are always headed/labeled with the time. This time can be logged individually for each item.

**All** items are always separated by a separator that the user selects in the GUI – see section 2.3.1. of this manual, **Edit Custom Log format**.

The data format is structured in lines with data items.

For a detailed description of the individual data record we refer to the following sections in the General format section: 4.1.2 E (Event record), 4.1.4 P (Position record), 4.1.5 A (Attitude record: RPH), 4.1.6 D (Data acquisition record), 4.1.7 G (Gyro record), 4.1.9 V (velocity record).

**Note**: Raw data cannot be saved in this format. Gyro records only hold the gyro value, not CMG.

The custom logfile includes two different time stamps:

1. The item called **Time**
   This represents a general time for the data recording record, and is not related to a specific field in the records
2. **Save Date/Time on a general item**
   This represents the timestamp of this specific field

# 5 References

You can find a total list of manual and documents on the EIVA documentation site. Access is granted via www.eiva.com

# 6 Figures