Tritech Genesis

Software Suite

Supported Interface Strings

and File Formats

0716-SOM-00002 Rev 02

Tritech

Table of Contents

Help & Support

First please read this manual thoroughly (particularly the Troubleshooting section, if present). If a warranty is applicable, further details can be found in the Warranty Statement, 0080-STF-00139, available upon request.

Tritech International Ltd can be contacted as follows:

	Mail	Tritech International Ltd Peregrine Road Westhill Business Park Westhill, Aberdeenshire AB32 6JL, UK
	Telephone	+44(0)1224 744 111
	Email	support@tritech.co.uk
	Website	www.tritech.co.uk

Prior to contacting Tritech International Ltd please ensure that the following is available:

1. The Serial Numbers of the product and any Tritech International Ltd equipment connected directly or indirectly to it.
2. Software or firmware revision numbers.
3. A clear fault description.
4. Details of any remedial action implemented.

Contamination
If a product has been used in a contaminated or hazardous environment you must de-contaminate the product and report any hazards prior to returning the unit for repair. Under no circumstances should a product be returned that is contaminated with radioactive material.

The name of the organisation which purchased the system is held on record at Tritech International Ltd and details of new software or hardware packages will be announced at regular intervals. This manual may not detail every aspect of operation and for the latest revision of the manual please refer to www.tritech.co.uk.

Tritech International Ltd can only undertake to provide software support of systems loaded with the software in accordance with the instructions given in this manual. It is the customer's responsibility to ensure the compatibility of any other package they choose to use.

Introduction

Genesis is Tritech International's most recent all in one software interface for controlling, displaying and recording data from its portfolio of sensors including multibeam & mechanical sonars, USBL positioning, bathymetric and echosounder devices. It can also be used to record data from cameras, GPS, MRU and other third party device connections.

It boasts a modern, dynamic user interface with highly integrated features and builds upon decades of experience in providing an easy to use user interface for the wide variety of subsea sensors manufactured by Tritech International Ltd. Devices can be dynamically added, configured and setup within the Genesis GUI.

Genesis is available for Windows OS® only. The Genesis software is under continuous further development to support new products, add new features and new functionality. Tritech welcome any suggestions for further functionality or features. These and any software bug reports can be sent to support@tritech.co.uk for consideration and / or resolution.

Software functionality and details are correct for version 1.7.3 of Genesis and may differ from earlier or later versions. The latest released version of Genesis is available from the Tritech website www.tritech.co.uk/.

This supplementary manual details the interface strings that are recognised by Genesis as valid inputs, and can be output from Genesis via a serial connection. Several of the output strings are only relevant to specific Tritech devices and may not be available for all other products. This document also documents the common text interface file formats for inputs and outputs from Genesis.

This manual should be used as a reference and consulted along with the relevant product specific hardware or software manual. The current release of this and other manuals are available from the Tritech website www.tritech.co.uk/.

Genesis Input & Output Data Strings

This section details the various input and output ASCII serial data string types supported by Genesis.

Attitude Heading Pitch & Roll Strings

The following Attitude Heading Pitch & Roll (AHRS) strings are accepted as inputs into Genesis.

Strings with Attitude

String	Description
TCM2	$C123.4P12.3R12.3T12.3X123.4Y123.4Z123.4E000*FF<CR><LF>
NMEA ‘TRO’	$--TRO,x.xx,a,y.yy,b*kk<CR><LF>
NMEA ‘TRH’	$--TRH,x.xx,a,y.yy,b,z.zz,c*kk<CR><LF>
CDL1	H123.4P+123.45R+123.45T00.0D00000.0B00.0FR<CR><LF>
MDL	H0750P-0019R-0022<CR><LF>
Digilog/Oceantools	H0750P-0019R-0022E<CR><LF>
CDL Microtilt	P+12.34R+12.34<CR><LF>
TSS1	:XXAAAASMHHHHQMRRRRSMPPPP<CR><LF>
SMC ‘SMCS’	$PSMCS,±yy.yyy,±xx.xxx,±hh.hh<CR><LF>
SMC ‘SMCD’	$PSMCD,±yy.yy,±xx.xx,±zzz,z,±yv.yv,±xv.xv,±zv.zv*hh<CR><LF>
VideoRay ‘PVRND’	$PVRND,mm/dd/yyyy,hh:mm:ss.s,d.d,h.h,p.p.r.r*cs<CR><LF>

Strings with Heading

String	Description
HDG	$--HDG,x.x,x.x,a,x.x,a*hh<CR><LF>
HDT	$--HDT,x.x,T*hh<CR><LF>
HDM	$--HDM,x.x,M*hh<CR><LF>

Global Positioning Strings

The following Global Positioning System (GPS) and other positional system strings can be used as both inputs and navigational output strings within Genesis.

NMEA GPGLL

This NMEA string format is used for Global Positioning and mainly sent by GPS devices. It is a common format for providing co-ordinate positions in Latitude / Longitude format. This ASCII string has comma field separators.

The full string format is as follows:

'$GPGLL' Reply Data Structure		
Data Description	**Data Range**	**Field Range**
NMEA string header	Always '$GPGLL'	hhhhhh
Latitude of Target in Degrees Min (Only valid when Compass and GPS input data are valid)	0 to 90 Degrees	DDMM.SSSS
Latitude N or S	'N' or 'S'	a
Longitude of Target in Degrees Min (Only valid when Compass and GPS input data are valid)	0 to 180 Degrees	DDDMM.SSSS
Longitude E or W	'E' or 'W'	a
Time of data, UTC = hhmmss.ss	000000.00 to 235959.99	hhmmss.ss
Fix Status (invalid if Ping not okay or no valid attitude or compass data)	'A' = Valid, 'V' = Invalid	x
Delimiter (asterisk)	Always '*'	*
Checksum (in Hexadecimal) (XOR of bytes between, but not inclusive of, '$' and '*')	00 to FF	cc
(Carriage Return + Line Feed)	N/A	<CR LF>

Example:

Target Latitude = 54º 22' 22.21" (N), Target Longitude = 2º 56' 23.91" (W), UTC Time of reading = 143231.51.

ASCII Output = "$GPGLL,5422.3701,N,00256.3986,W,143231.51,A*54<CR><LF>"

Where; "<CR>" is Carriage Return (ASCII code 13), "<LF>" is Line Feed (ASCII code 10).

NMEA GPGGA

This NMEA string format is used for Global Positioning and mainly sent by GPS devices. It is a common format for providing co-ordinate positions in Latitude / Longitude format. When used as an output from Genesis there are a number of fields in this string that are not applicable and these fields have been set to sensible values as default. This ASCII string has comma field separators.

The full string format is as follows:

'$GPGGA' Reply Data Structure		
Data Description	**Data Range**	**Field Range**
NMEA string header	Always '$GPGGA'	hhhhhh
Time of data, UTC = hhmmss.ss	000000.00 to 235959.99	hhmmss.ss
Latitude of Target in Degrees Min (Only valid when Compass and GPS input data are valid)	0 to 90 Degrees	DDMM.SSSS
Latitude N or S	'N' or 'S'	a
Longitude of Target in Degrees Min (Only valid when Compass and GPS input data are valid)	0 to 180 Degrees	DDDMM.SSSS
Longitude E or W	'E' or 'W'	a
GPS Quality Indicator	Always 2 (DGPS)	x
Number of Satellites in Use	Always 7 (N/A)	xx
HDOP	Always 2.2	x.x
Antenna Altitude	Always 0.0	x.x
Altitude Units Identifier	'M' for metres	a
Geoidal Separation	Always 0.0	x.x
Units Identifier	'M' for metres	a
Age of Differential GPS data	Always 1.2	x.x
Differential Reference Station ID	Always 1234	xxxx
Delimiter (asterisk)	Always '*'	*
Checksum (in Hexadecimal) (XOR of bytes between, but not inclusive of, '$' and '*')	00 to FF	cc
(Carriage Return + Line Feed)	N/A	<CR><LF>

Example:

UTC Time of reading = 144014.71, Target Latitude = 54º 22' 21.74" (N), Target Longitude = 2º 56' 25.07" (W).

ASCII Output Always =
"$GPGGA,144014.71,5422.3624,N,00256.4178,W,2,07,2.2,0.0,M,0.0,M,1.2,1234*5C<CR><LF>"

Where; "<CR>" is Carriage Return (ASCII code 13), "<LF>" is Line Feed (ASCII code 10).

NMEA GPRMC

This NMEA string format is used for Global Positioning and mainly sent by GPS devices. It is a common format for providing co-ordinate positions in Latitude / Longitude format. This ASCII string has comma field separators.

The full string format is as follows:

'$GPRMC' Reply Data Structure		
Data Description	**Data Range**	**Field Range**
NMEA string header	Always '$GPRMC'	hhhhhh
Time of data, UTC = hhmmss.ss	000000.00 to 235959.99	hhmmss.ss
Fix Status (N.B. Invalid if Ping not okay or no valid attitude or compass data)	'A' = Valid, 'V' = Invalid	x
Latitude of Target in Degrees Min (Only valid when Compass and GPS input data are valid)	0 to 90 Degrees	DDMM.SSSS
Latitude N or S	'N' or 'S'	a
Longitude of Target in Degrees Min (Only valid when Compass and GPS input data are valid)	0 to 180 Degrees	DDDMM.SSSS
Longitude E or W	'E' or 'W'	a
Speed Over Ground (in knots)	Not calculated, always 0.00	x.xx
Course Over Ground (in degrees)	Not calculated, always 000.00	xxx.xx
Date of data	Gregorian calendar 010100 to 311299	ddmmyy
Magnetic Variation	N/A, always ''	x
Delimiter (asterisk)	Always '*'	*
Checksum (in Hexadecimal) (XOR of bytes between, but not inclusive of, '$' and '*')	00 to FF	cc
(Carriage Return + Line Feed)	N/A	<CR LF>

Example:

Target Latitude = 54º 22' 22.30" (N), Target Longitude = 2º 56' 25.51" (W), UTC Time of reading = 143221.57, Date of reading = 9th May 2008.

ASCII Output Always =
"$GPRMC,143221.57,A,5422.3717,N,00256.4252,W, 0.00,000.00,090508,,*0F<CR><LF>"

Where; "<CR>" is Carriage Return (ASCII code 13), "<LF>" is Line Feed (ASCII code 10).

NMEA RATTM

This NMEA 'Tracked Target Message' is a commonly used string in charting/navigation packages. The comma-separated ASCII string contains the target Range & Bearing and UTC Time-stamp. When used as an output from Genesis the target speed and course are not calculated and so are not contained in the string. Any fields with no calculated data will be padded with zero value data.

The full string format is as follows:

'$RATTM' Reply Data Structure		
Data Description	**Data Range**	**Field Range**
NMEA string header	Always '$RATTM'	hhhhhh
Target number	00 to 99	ii
Horizontal Target Distance from vessel (in nautical miles)	0 to 9.9999	d.dddd
Target Bearing (degrees)	0 to 359.9	bbb.b
Bearing Degrees identifier	'R' or 'T'	'R' (Relative) or 'T' (True)
Target Speed, not calculated so padded with zero	Always '0.0'	s.s
Degrees identifier, unused so defaulted to 'T'	'R' or 'T'	'R' (Relative) or 'T' (True)
Distance of closest point of approach, not calculated so padded with zero	Always '0.0'	p.p
Time to CPA, min, ("-" increasing), not calculated so padded with zero	Always '0.0'	t.t
Speed/Distance units	Always 'N' (nautical miles)	K/N/S (km/ nautical miles/ statue miles)
Target name (0 = Responder, 1..16 = Transponder)	e.g., "TARGET0" to "TARGET1"	ASCII variable field
Target Status	'L' or 'T'	L = Lost, Q = Query in process, T = Tracking
Target ?	Always null	R = Reference Target, otherwise null
Time of data, UTC = hhmmss.ss	000000.00 to 235959.99	hhmmss.ss
Type of acquisition	Always 'A'	A = Auto, M = Manual, R = Reported

Delimiter (asterisk)	Always ‘*’	*
Checksum (in Hexadecimal) (XOR of bytes between, but not inclusive of, '$' and '*')	00 to FF	cc
(Carriage Return + Line Feed)	N/A	<CR><LF>

Example:

Target Number / Unit ID = 0 (Responder), Horizontal Target Range = 2 metres = 0.0011 nautical miles, Relative Target Bearing = 326.0º, UTC Time of reading = 005940.13.

ASCII Output Always =
“$RATTM,00,0.0011,326.0,R,0.0,0.0,R,0.0,0.0,N, TARGET1,T,,005940.13,A*5B<CR><LF>”

Where; “<CR>” is Carriage Return (ASCII code 13), “<LF>” is Line Feed (ASCII code 10).

Simrad HiPAP SSB (PSIMSSB)

This 3rd party output string is taken from the Simrad HiPAP positioning system. The string is variable length ASCII with comma field separators. The X, Y, Z co-ordinate positions are relative values, unless the GPS is active, in which case the co-ordinate outputs become Easting, Northing and Depth, respectively.

The full string format is as follows:

'$PSIMSSB' Reply Data Structure		
Data Description	**Data Range**	**Field Range**
NMEA string header	Always '$PSIMSSB'	hhhhhhhh
Time of data (UTC time)	000000.00 to 235959.99	hhmmss.ss
Beacon/Target ID (0 = Responder, 1..16 = Transponder	B00 to B99	ccc
Target Status	'A' =Okay, 'V' = Not Okay	A
Error Code, left empty	Always blank	
Coordinate system	'C' = Relative Cartesian, 'U' = Absolute UTM	A
Orientation	'H' for Heading up & 'N' for North up (both when output is Relative XY), 'E' for Easting, Northing (when in Absolute UTM format)	A
SW Filter	'M' for Measured, 'F' for Filtered	A
X (or Easting)	0 to 999999	x.x
Y (or Northing)	0 to 999999	y.y
Z (or Depth)	0 to 999999	z.z
Expected Accuracy, unused	Always 0.0	q.q
Additional information	Always 'N'	A
1st Additional value	Always blank	
2nd Additional value	Always blank	

Delimiter (asterisk)	Always '*'	*
Checksum (in Hexadecimal) (XOR of bytes between, but not inclusive of, '$' and '*')	00 to FF	cc
(Carriage Return + Line Feed)	N/A	<CR LF>

Example:

GPS is active so co-ordinates are World. Time of reading = 204854.17, Beacon ID = 00 (Responder), X Co-ord (East) of 538087.76, Y Co-ord (North) of 7039253.894, Z Co-ord (Depth) of 56.486m, Status = Okay.

ASCII Output Always =
"$PSIMSSB,204854.17,00,A,,C,H,M,538087.760, 7039253.894,56.486,0.0,N,,*26<CR><LF>"

Where; "<CR>" is Carriage Return (ASCII code 13), "<LF>" is Line Feed (ASCII code 10).

TrackPoint TP-2EC

This 3rd party output string is taken from the TrackPoint positioning system. The string has 68 characters, and the format is ASCII fixed field with space separators. Leading 0's (except time) are space filled. X, Y, Z co-ordinate positions are relative values, unless GPS is active in which case the co-ordinate outputs become Easting, Northing and Depth respectively.

The full string format is as follows:

'TP-2EC' Reply Data Structure		
Data Description	**Data Range**	**Data Types**
Unit ID, in Hex (0 = Responder, 1..F = Transponder	0 to Fh (0 to 19)	i
Fix Time (Local Time)	00:00:00 to 23:59:59	hh:mm:ss
Compass Heading (degrees)	0 to 359	ccc
Position Bearing (degrees)	0 to 359.9	bbb.b
Position Slant Range (metres)	0 to 99999.9	rrrrr.r
X (or Easting)	0 to 999999.9	xxxxxx.x
Y (or Northing)	0 to 999999.9	yyyyyy.y
Z (or Depth)	0 to 99999.9	zzzzz.z
Telemetry	Always 0.0	t.t
Error	' '(No Error) or ' 6' (Lost Signal)	ee
(Carriage Return + Line Feed)	N/A	<CR LF>

Example:

GPS is active so co-ordinates are World. Unit ID = 0 (Responder), Fix taken at 15:33:02, Compass Heading = 337º, Position Bearing = 277.6º, Position Slant Range = 20.8m, X Co-ord (East) of 503871.9, Y Co-ord (North) of 602501.0, Z Co-ord (Depth) of 13.0m, No Error.

ASCII Output Always =
"0*15:33:02*337*277.6****20.8*503871.9* 602501.0****13.0******0.0***<CR><LF>"

Where; "*" is a space (ASCII code 32), "<CR>" is Carriage Return (ASCII code 13), "<LF>" is Line Feed (ASCII code 10).

Simrad HPR 300P

This 3rd party output string is taken from the Simrad HPR positioning system. The string has 48 characters, and the format is ASCII fixed field with space separators. Leading 0's (except time) are space filled. X, Y, Z co-ordinate positions are relative values, unless GPS is active in which case the co-ordinate outputs become Easting, Northing and Depth respectively. For values greater than 999.9, the decimal point is removed (e.g., 1001.1 is output as "1001").

The full string format is as follows:

'$HPR 300P' Reply Data Structure		
Data Description	**Data Range**	**Field Range**
Beacon / Unit ID (0 = Responder, 1..16 = Transponder	0 to 19	ii
Transducer Number	Always 1	H
Beam	Always 'W'	B
Status	' OK' (No error) or 'NRY' (No reply)	SSS
Compass heading (degrees)	0 to 359.9	ccc.c
X (or Easting)	0 to 999999	xxxx.x (or xxxxxx)
Y (or Northing)	0 to 999999	yyyy.y (or yyyyyy)
Z (or Depth)	0 to 999999	zzzz.z (or zzzzzz)
Quality Flag, unused	Always 0.0	QQ.Q
(Carriage Return + Line Feed)	N/A	<CR><LF>

Example:

GPS is active so co-ordinates are World. Unit ID = 0 (Responder), Compass Heading = 119.5º,X Co-ord (East) of 339080, Y Co-ord (North) of 497512, Z Co-ord (Depth) of 13.7m, No Error.

ASCII Output Always = "0**1**W***OK**119.5*339080*497512***13.7***0.0<CR><LF>"

Where; "*" is a space (ASCII code 32), "<CR>" is Carriage Return (ASCII code 13),"<LF>" is Line Feed (ASCII code 10).

Navigational Positioning Strings

In addition to the GPS and positional strings, the following strings can be used to output positional and tracking information from Genesis when used with a MicronNav system.

Proc XYZ

The 'Proc XYZ' data message contains the World X, Y & Z co-ordinate position of the Responder/Transponder. The output message will include a message header and be in the following format, in accordance with the Tritech proprietary Input/ Output protocol.

"%D" + SlotReplyHdr + 'Proc XYZ' Nav Data Structure + <CR> <LF>

SlotReplyHdr Data Structure (sent in hexadecimal format)		
Data Description	**Data Range**	**Data Types**
Total Number of Bytes in Message in Hex (including Command and Reply codes)	NB	CARDINAL
Slot Number (range "01" to "0C")	SlotN	SLOTN
Generic Device Type	SourceTypes	SOURCEN
Data Reply Mode (0=ASCIIText, 1=Hex, 2=Binary, 3=CSV)	0,1,2 or 3	DIGIT
Send SeaKing Long = 3*, Send SeaKing Short = 2*, Send Raw data = 1, Send Processed Data = 0 (*Not applicable)	0,1,2 or 3	DIGIT

Example:

Byte Count = Hex 0049 (73)
Slot = 08 = MicronNav
Source type = 32(Hex 20) = Null (not defined for MicronNav)
Data reply mode is ASCIIText
Send data = 0 = Proc XYZ
ALWAYS Hex e.g., "0049082000"

‘Proc XYZ’ Nav Data Structure		
Data Description	**Data Range**	**Data Types**
Unit ID (0 = Responder, 1..16 = Transponder)	0 to 19	SHORTCARD
World X co-ordinate (Easting), including datum position offset.	-9.99999E-37 to +9.99999E+37	REAL
World Y co-ordinate (Northing), including datum position offset.	-9.99999E-37 to +9.99999E+37	REAL
Vertical position relative to vessel transducer and including datum position offset (in millimetres, +ve is downwards)	0000000000 to 100000000	LONGINT
Quality Flag / RMS Error. Larger values indicate better quality.	0.0 to 3.0	REAL
Valid Reply Set Bit 0 = 1 = Tx Transducer Ok Bit 1 = 1 = Rx 1 Transducer Reply Ok Bit 2 = 1 = Rx 2 Transducer Reply Ok Bit 3 = 1 = Rx 3 Transducer Reply Ok Bit 4 = 1 = Rx 4 Transducer Reply Ok e.g., Valid reply = Bits 1 to 4 set to 1 = “30” (or "31")	000 to 031	SHORTCARD
Time of data, local time = hhmmssdd	00000000 to 23595999	hhmmssdd

Example:

Unit ID = 0 (Responder), World X co-ordinate (Easting) of 503868.427, World Y co-ordinate (Northing) of 6025011.669,Vertical Z co-ordinate of 4.102m, Quality Flag = 0.8, Valid reply Set = 30 (all valid), Fix taken at 15:31:56.

ASCII Text = "%D004B082000000+5.03868427E+05+6.02501167E+06+0000004102 +0.80000E+0003015315632<CR><LF>”

CSV = "%D003E082030,0,503868.427,6025011.669,+4102,0.8,30,15315632<CR><LF>”

Raw XYZ

The ‘Raw XYZ’ data message contains the Relative X, Y & Z co-ordinate position of the Responder/Transponder. The output message will include a message header and be in the following format, in accordance with the Tritech proprietary RemV4 Output protocol

“%D” + SlotReplyHdr + ‘Raw XYZ’ Nav Data Structure + <CR> <LF>

SlotReplyHdr Data Structure (this Data Structure is sent in Hex Format)		
Data Description	**Data Range**	**Data Types**
Total Number of Bytes in Message in Hex (including Command and Reply codes)	NB	CARDINAL
Slot Number (range “01” to “0C”)	SlotN	SLOTN
Generic Device Type	SourceTypes	SOURCEN
Data Reply Mode (0=ASCIIText, 1=Hex, 2=Binary, 3=CSV) *CSV = Comma Separated ASCII	0 or 1 or 2 or 3	DIGIT
Send SeaKing Long = 3*, Send SeaKing Short = 2*, Send Raw data = 1, Send Processed Data = 0 *Not applicable	0 or 1 or 2 or 3	DIGIT
Example: Byte Count = Hex 0049 (73) Slot = 08 = MicronNav Sourcetype = 32(Hex 20) = Null (not defined for MicronNav) Data reply mode is ASCIIText Send data = 1 = Raw XYZ ALWAYS Hex e.g., “0049082001”		

‘Raw XYZ’ Nav Data Structure (this Data Structure is sent in Hex Format)		
Data Description	**Data Range**	**Data Types**
Unit ID (0 = Responder, 1..16 = Transponder)	0 to 19	SHORTCARD
Relative X co-ordinate. Horizontal distance athwart from reference position (in millimetres, +ve = towards starboard)	0000000000 to 100000000	LONGINT
Relative Y co-ordinate. Horizontal distance fore/aft from reference position (in millimetres, +ve = forwards)	0000000000 to 100000000	LONGINT
Relative Z co-ordinate. Vertical distance from reference position (in millimetres, +ve is downwards)	0000000000 to 100000000	LONGINT
Quality Flag / RMS Error. Larger values indicate better quality.	0.0 to 3.0	REAL
Valid Reply Set Bit 0 = 1 = Tx Transducer Ok Bit 1 = 1 = Rx 1 Transducer Reply Ok Bit 2 = 1 = Rx 2 Transducer Reply Ok Bit 3 = 1 = Rx 3 Transducer Reply Ok Bit 4 = 1 = Rx 4 Transducer Reply Ok E.G. Valid reply = Bits 1 to 4 set to 1 = “30” (or "31")	000 to 031	SHORTCARD
Time of data, local time = hhmmssdd	00000000 to 23595999	hhmmssdd

Example:

Unit ID = 0 (Responder), Relative X co-ord of -18.686m, Relative Y co-ord of 4.764m, Relative Z co-ord of 4.1m, Quality Flag = 0.8, Valid reply Set = 30 (all valid), Fix taken at 15:31:56.

ASCII Text = “%D0049082001000-0000018686+0000004764+0000004100+0.80000E +0003015315632<CR><LF>”

CSV =“%D0033082031,0,-18686,+4764,+4100,0.8,30,15315632<CR><LF>”

Simrad HPR 410

This 3rd party output string is taken from the Simrad HPR positioning system. The binary output telegram is 70 bytes long. The binary telegram includes a Telegram Header and Tail in line with the Simrad HPR protocol. The contained Message Type is 'Message 1, Transponder Position Data' (Data Block of 62 bytes).

The full binary telegram format is as follows:

'$HPR 410' Reply Data Structure		
Data Description	**Data Range**	**Field Range**
Header		
Start Character	Always 55h	ShortCard
Block Length	1 to 65536	Cardinal
Message Type	Always 1	ShortCard
Destination	Always 0	ShortCard
Message Block (62 bytes)		
Tp Index / ID (0 = Responder, 1..16 = Transponder)	0 to 19	Cardinal
Operation Mode	Always 0	ShortCard
Sync Mode	Always 0	ShortCard
Tp Mode/Type	0 = Transponder, 10 = Responder	ShortCard
Tp Operation	Always 1 (Mobile Tp)	ShortCard
Pos Data Form	Bit 0 (0 = vessel, 1 = north)	ShortCard
Reply Status	Fixed at 0 (= Okay)	ShortCard
Filt x pos - Filtered Horiz Distance athwart from reference pos (in metres)	-9.99999E-37 to +9.99999E+37	Real
Filt y pos - Filtered Horiz Distance fore/aft from reference pos (in metres)	-9.99999E-37 to +9.99999E+37	Real
Filt z pos - Filtered Vert Distance from reference pos (in metres)	-9.99999E-37 to +9.99999E+37	Real
X pos - Raw Horiz Distance athwart from reference pos (in metres)	-9.99999E-37 to +9.99999E+37	Real
Y pos - Raw Horiz Distance fore/aft from reference pos (in metres)	-9.99999E-37 to +9.99999E+37	Real

‘$HPR 410’ Reply Data Structure (continued)		
Data Description	**Data Range**	**Field Range**
Z pos - Raw Vert Distance from reference pos (in metres)	-9.99999E-37 to +9.99999E+37	Real
Slant Range - Raw Slant Range from transducer to Transponder (in metres)	-9.99999E-37 to +9.99999E+37	Real
P course - Vessel heading Course (in degrees)	-9.99999E-37 to +9.99999E+37	Real
P roll - Vessel Roll (+/-180 degrees)	-9.99999E-37 to +9.99999E+37	Real
P pitch - Vessel Pitch (+/-180 degrees)	-9.99999E-37 to +9.99999E+37	Real
Td beam	0 = wide, 1 = narrow	ShortCard
Td type	Not used, Always 0	ShortCard
Td num	Always 1	Cardinal
Diagnostic	Not used, Always 0	Cardinal
Standard deviation, unused	Always 0.0	Real
Inst Data	Not used, Always 0	Real
Footer Tail		
Checksum(Sum of all bytes excluding Checksum and Stop Character)	1 to 65536	Cardinal
Stop Character	Always AAh	ShortCard
Example: The output telegram is binary so a viewable example is not given.		

NMEA GPDBT

This NMEA string format is used for Water Depth data output and is the Depth Below Transducer value that is mainly sent by positioning devices. It provides the Z (Depth) coordinate output position in the USBL system. This ASCII string has comma field separators.

The full string format is as follows:

'$GPDBT' Reply Data Structure		
Data Description	**Data Range**	**Field Range**
NMEA string header	Always '$GPDBT'	hhhhhh
Depth in units of Feet	0 to 999999	f.ff
Units Identifier	Always 'f'	U
Depth in units of Metres	0 to 999999	M.MM
Units Identifier	Always 'M'	U
Depth in units of Fathoms	0 to 999999	F.FF
Units Identifier	Always 'F'	U
Delimiter (asterisk)	Always '*'	*
Checksum (in Hexadecimal) (XOR of bytes between, but not inclusive of, '$' and '*')	00 to FF	cc
(Carriage Return + Line Feed)	N/A	<CR LF>

Example:

Z (Depth) value = 54.12 metres.

ASCII Output Always = "$GPDBT,177.55,f,54.12,M,29.59,F*32<CR><LF>"

Where; "<CR>" is Carriage Return (ASCII code 13), "<LF>" is Line Feed (ASCII code 10).

Bathymetric Sensor Data Strings

The data strings relevant to the Bathymetric Sensor are the same as those available within Seanet Pro. Details of the data strings can be found in manual 0706-SOM-00004 Seanet Remote Communications.

String Data Type Formats

I/O Data Types			
Data Type	**Binary Mode**	**Hex Mode**	**ASCII Text Mode**
REMCH			":"
REPCH			"%"
BOOLEAN	Nibble	"b"	"0" or "1"
DIGIT	Nibble	"n"	Any Digit "0" to "9"
CHAR	<byte>	<byte>	Any printable ASCII
SHORTCARD	<byte>	"Nn"	"000" to "255"
SHORTINT	<byte>	"Nn"	"-128" to "+128"
CARDINAL	<LSB><MSB>	"MmLl"	"00000" to "65535"
INTEGER	<LSB><MSB>	"MmLl"	"-32768" to "+32767"
LONGCARD	<LSB><.><.><MSB>	"Mm....Ll"	"0000000000" to "4294967296"
LONGINT	<LSB><.><.><MSB>	"Mm....Ll"	"-2147483648" to "+2147483647"
REAL	<LSB><.><.><MSB>	"Mm....Ll"	"-9.99999E-37" to "+9.99999E+37"
LONGREAL	<LSB>,6*<><MSB>	"Mm.....Ll"	"-9.99999999E-307" to "+9.99999999E+307"
TIME	<C><S><M><H>	"HhMmScCc"	"HHMMSSCC"
DATE	<D><M><Y>	"DdMmYyyy"	"DDMMYYYY"
SLOTN	<1..12>	"Nn" ("01" to "0C")	"01" to "12"
SOURCEN	<0..99>	"Nn" ("00" to "63")	"00" to "99"
DEVICEN	<0..99>	"Nn" ("00" to "63")	"00" to "99"
NODEN	<1..16>	"Nn" ("01" to "63")	"01" to "99"

Genesis Input & Output Text Data Files

Genesis Chart Marker File Formats

These file formats contain the data to load and save the details of any Chart marker points created (or loaded) into Genesis or SeaNet Pro.

Full Marker File Format (.mrk extension)

This is the format that is native to Genesis and Seanet Pro and includes full details of the marker configuration. The file is in an ASCII Comma Separated format and contains the following fields:

Index	Value	Description
1	ID	This is a unique string ID. It comprises a 2-letter header ("mk") followed by a Date Time code. Any unique string value is acceptable.
2	Group	This is unused and should be set to 0.
3	X Coordinate	For the .mrk file this will always be output in UTM Easting.
4	Y Coordinate	For the .mrk file this will always be output in UTM Northing.
5	Altitude	This is UTM Altitude and is currently unused.
6	UTM Zone Parallel	Zone latitudinal letter (e.g., 'C' through 'X').
7	UTM Zone Meridian	Zone longitudinal number (e.g., 1 through 60).
8	UTM Ellipsoid	Ellipsoid code (0 = Airy, 1 = Australian National, 2 = Bessel1841, 3 = Clarke 1866, 4 = Clarke 1880, 5 = Everest, 6 = GRS80, 7 = International 1924, 8 = Modified Airy, 9 = WGS84).
9	Point Size	Applies to Circle, Square & Triangle shape types only, otherwise set to 0.
10	Date & Time	Date & Time in English (GB) Locale. Format is "dd/mm/yyyy hh:mm:ss"
11	Shape Type	0 = Circle, 1 = Square, 2 = Triangle, 3 = Sonar Range, 4 = Pre-set Image (see Image Info below).
12	Shape Colour	Applies to Circle, Square and Triangle shape types only, otherwise set to 00000000 (32-bit RGBA).
13	Font Inner Colour	Applies to Comment Text (32-bit RGBA).
14	Font Outer Colour	Applies to Comment Text (32-bit RGBA).
15	Marker Bitwise	Bit 1 = Show Marker, Bit 2 = Show Coordinates, Bit 3 = Show Comment Text (i.e., 00000111 = Show All).
16	Image Info	Pre-sets = Red Flag, Blue Flag, Green Flag, Buoy, Anchor, Rock, Danger, POI, Viewport, Sonar, Diver, Wheel, Comment or MLO Alternatively can be full path and name of an image file (e.g., 'C: \Image1.bmp').
17	Comment	Comment text.

Shortened Marker File Format (.csv extension)

This file format is a more concise and usable format, particularly for loading a pre-defined marker list into Genesis. The file is in an ASCII Comma Separated format and contains the following fields:

Index	Value	Description
1	ID	This is a unique string ID. It comprises a 2-letter header ("mk") followed by a Date Time code. Any unique string value is acceptable.
2	X Coordinate	For the .csv file, will be in coordinate system used in Genesis (either Longitude or UTM Easting).
3	Y Coordinate	For the .csv file, will be in coordinate system used in Genesis (either Latitude or UTM Northing).
4	Comment	Comment text.
5	Shape Type	0 = Circle, 1 = Square, 2 = Triangle, 3 = Sonar Range, 4 = Pre-set Image (see Image Info below).
6	Image Info	Pre-sets = Red Flag, Blue Flag, Green Flag, Buoy, Anchor, Rock, Danger, POI, Viewport, Sonar, Diver, Wheel, Comment or MLO Alternatively can be full path and name of an image file (e.g., 'C:\Image1.bmp').
7	Date & Time	Date & Time in English (GB) Locale. Format is "dd/mm/yyyy hh:mm:ss"

Shortened .csv File Variations

All field data
All 7 fields contain data as normally created when Genesis saves the .csv file.

First 4 fields only
There is no entry for Shape, Image Info or Date & Time. Since no shape data is available all the markers will use a red flag.

First 5 or 6 fields only
No Date & Time entry, also, if the Shape Type is 0, 1, 2 or 3 then the Image Info entry can be omitted (since there will be no need for it).

Bathymetric Sensor Profile (.BP3) Data File

During descent, calculations and measurements taken by the Bathy unit are stored in a look-up table at index intervals of 1psi. This is for the purpose of maintaining a running calculation of mean density and mean velocity of sound during deployment down to operating depth.

The look up table is a space-delimited text file (with file extension .BP3) that can be imported into a spreadsheet for further analysis or post-processing. Each column of the file is space delimited. The first column (containing the Index Number) has a fixed width of 14 characters and every other column thereafter has a fixed width of 20 characters.

The first line of the text file identifies the columns and also indicates the number of entries in the file as a number in the first column, for example, a file with 2979 entries would have this as a first line:

```
BP3 2979    Loc Dens    Mean Dens   Pressure    Temp        Salinity
```

```
Loc VOS     Mean VOS    Depth
```

It is very important to remember to save and therefore close off a recorded profile. Depending on the length of the descent to the working depth, a profile can take several seconds to be saved to disk.