SPAN® on OEM6®
Firmware Reference Manual

OM-20000144 Rev 8 December 2016

SPAN on OEM6 Firmware Reference Manual

Publication Number:	OM-20000144
Revision Level:	8
Revision Date:	December 2016

This manual reflects firmware version 6.63 / OEM060630RN0000.

Proprietary Notice

Table of Contents

Figures

Tables

Customer Support

NovAtel Knowledge Base

If a technical issue is encountered, browse to the NovAtel Web site at www.novatel.com/support. Use this page to search for general information about GNSS and other technologies, information about NovAtel hardware and software and installation and operation issues.

Before Contacting Customer Support

Before contacting NovAtel Customer Support about a software problem perform the following steps:

1. Log the following data to a file on your computer for 15 minutes.

```
RXSTATUSB once
RAWEPHEMB onchanged
GLORAWEPHEMB onchanged
RANGECMPB ontime 1
BESTPOSB ontime 1
RXCONFIGA once
VERSIONA once
RAWIMUSXB onnew
INSPVAXB ontime 1
INSUPDATEB onchanged
IMUTOANTOFFSETSB onchanged
VEHICLEBODYROTATIONA once
```

2. Send the file containing the log to NovAtel Customer Support, using either the NovAtel FTP site at ftp://anonymous@ftp.novatel.com/incoming or through the support@novatel.com e-mail address.
   If a file is sent using the FTP site, please contact Customer Support using one of the methods below.
3. A FRESET command can also be issued to the receiver to clear any unknown settings.

> The FRESET command will erase all user settings. You should know your configuration and be able to reconfigure the receiver before you send the FRESET command.

If a hardware problem is encountered, send a list of the troubleshooting steps taken and results.

Contact Information

Log a support request with NovAtel Customer Support using one of the following methods:

Log a Case and Search Knowledge:
Website: www.novatel.com/support

Log a Case, Search Knowledge and View Your Case History: (login access required)
Web Portal: https://novatelsupport.force.com/community/login

E-mail:
support@novatel.com

Telephone:
U.S. and Canada: 1-800-NOVATEL (1-800-668-2835)
International: +1-403-295-4900

Chapter 1 Introduction

NovAtel's SPAN technology brings together two different but complementary positioning and navigation systems: Global Navigation Satellite System (GNSS) and Inertial Navigation System (INS). By combining the best aspects of GNSS and INS into one system, SPAN technology offers a solution that is more accurate and reliable than either GNSS or INS alone. The combined GNSS/INS solution has the advantage of the absolute accuracy available from GNSS and the continuity of INS through traditionally difficult GNSS conditions.

1.1 About this manual

The SPAN receiver utilizes a comprehensive user-interface command structure, which requires communication through its communications ports.This manual describes the SPAN commands and logs the OEM6 family of receivers are capable of accepting or generating. Sufficient detail is provided to understand the purpose, syntax and structure of each command or log.

1.2 Related Documents and Information

This manual describes the SPAN specific commands and logs. For descriptions of the other commands and logs available with OEM6 family products, refer to the OEM6 Family Firmware Reference Manual (OM-20000129) available on the NovAtel website (www.novatel.com/support/manuals/).

For information about the installation and operation of the SPAN system, refer to the relevant SPAN User Manual available on the NovAtel website (www.novatel.com/support/manuals/).

- SPAN on OEM6 User Manual (OM-20000139)
- SPAN-CPT User Manual (OM-20000122) (Rev 7 or greater)
- SPAN-IGM User Manual (OM-20000141)

SPAN system output is compatible with post-processing software from NovAtel's Waypoint® Products Group. Visit our web site at www.novatel.com/products/software/ for details.

1.3 Conventions

The following conventions are used in this manual:

ⓘ	Information that supplements or clarifies text.
▼	A caution that actions, operation or configuration may lead to incorrect or improper use of the hardware.
⬣	A warning that actions, operation or configuration may result in regulatory noncompliance, safety issues or equipment damage.

Chapter 2 Messages

2.1 Message Types

The receiver handles incoming and outgoing NovAtel data in three different message formats: Abbreviated ASCII, ASCII and binary. This allows for a great deal of versatility in the way the OEM6 family of receivers can be used. All NovAtel commands and logs can be entered, transmitted, output or received in any of the three formats. The receiver also supports RTCA, RTCMV3, RTCM, CMR, CMRPLUS, NOVATELX and NMEA format messaging.

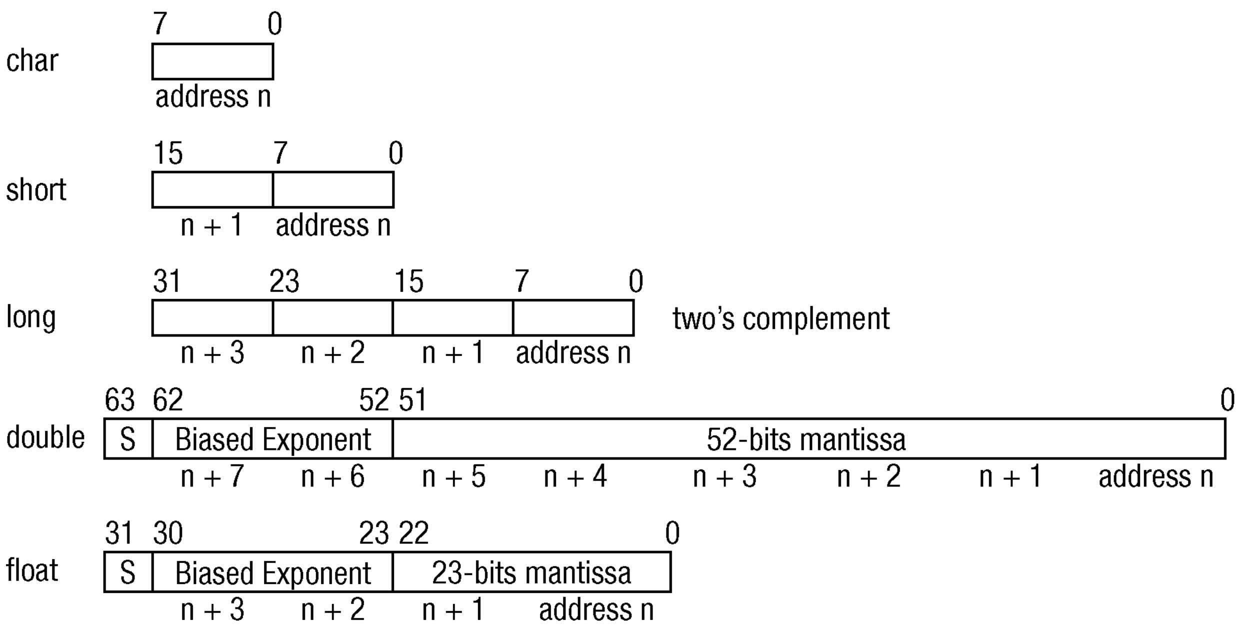
When entering an ASCII or abbreviated ASCII command to request an output log, the message type is indicated by the character appended to the end of the message name. 'A' indicates the message is ASCII and 'B' indicates binary. No character means the message is Abbreviated ASCII. When issuing binary commands, the output message type is dependent on the bit format in the message's binary header (refer to *Binary* on page 14).

Table 1, Field Types describes the field types used in the description of messages.

Table 1: Field Types

Type	Binary Size (bytes)	Description
Char	1	The char type is an 8-bit integer in the range -128 to +127. This integer value may be the ASCII code corresponding to the specified character. In ASCII or Abbreviated ASCII this comes out as an actual character
UChar	1	The uchar type is an 8-bit unsigned integer. Values are in the range from +0 to +255. In ASCII or Abbreviated ASCII this comes out as a number
Short	2	The short type is 16-bit integer in the range -32768 to +32767
UShort	2	The same as short except it is not signed. Values are in the range from +0 to +65535
Long	4	The long type is 32-bit integer in the range -2147483648 to +2147483647
ULong	4	The same as long except it is not signed. Values are in the range from +0 to +4294967295
Double	8	The double type contains 64-bits: 1 for sign, 11 for the exponent and 52 for the mantissa. Its range is ±1.7E308 with at least 15 digits of precision. This is IEEE 754
Float	4	The float type contains 32-bits: 1 for the sign, 8 for the exponent and 23 for the mantissa. Its range is ±3.4E38 with at least 7 digits of precision. This is IEEE 754
Enum	4	A 4-byte enumerated type beginning at zero (an unsigned long). In binary, the enumerated value is output. In ASCII or Abbreviated ASCII, the enumeration label is spelled out
GPSec	4	This type has two separate formats dependent on whether you requested a binary or an ASCII format output. For binary, the output is in milliseconds and is a long type. For ASCII, the output is in seconds and is a float type
Hex	n	Hex is a packed, fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs
Hex Ulong	4	An unsigned, 32-bit integer in hexadecimal format. Values are in the range from +0 to +4294967295
String	n	String is a variable length array of bytes that is null-terminated in the binary case and additional bytes of padding are added to maintain 4-byte alignment. The maximum byte length for each String field is shown in the row in the log or command tables

Figure 1: Byte Arrangements



> *Figure 1, Byte Arrangements* shows the arrangement of bytes, within each field type, when used by PC computers. All data sent to or from the OEM6 family of receivers, however, is read least significant bit (LSB) first, opposite to what is shown in *Figure 1, Byte Arrangements*. Data is then stored in the receiver LSB first. For example, in char type data, the LSB is bit 0 and the most significant bit (MSB) is bit 7.

2.1.1 ASCII

ASCII messages are readable by both the user and a computer. The structures of all ASCII messages follow the general conventions as noted here:

1. The lead code identifier for each record is '#'.
2. Each log or command is of variable length depending on amount of data and formats.
3. All data fields are delimited by a comma ',' with two exceptions:
   - The first exception is the last header field which is followed by a ';' to denote the start of the data message.
   - The second exception is the last data field, which is followed by a * to indicate end of message data.
4. Each log ends with a hexadecimal number preceded by an asterisk and followed by a line termination using the carriage return and line feed characters.
   For example:
   *1234ABCD[CR][LF]. This value is a 32-bit CRC of all bytes in the log, excluding the '#' identifier and the asterisk preceding the four checksum digits.
   See *32-Bit CRC* on page 25 for the algorithm used to generate the CRC.
5. An ASCII string is one field and is surrounded by double quotation marks.
   For example:
   "ASCII string". If separators are surrounded by quotation marks then the string is still one field and the separator will be ignored (example, "xxx,xxx" is one field). Double quotation marks within a string are not allowed.
6. If the receiver detects an error parsing an input message, it returns an error response message. See *Chapter 3, Responses* on page 27 for a list of response messages from the receiver.

Message Structure:

header;	data field...,	data field...,	data field...	*xxxxxxxx	[CR][LF]

The ASCII message header structure is described in *Table 2, ASCII Message Header Structure*.

Table 2: ASCII Message Header Structure

Field	Field Name	Field Type	Description	Ignored on Input
1	Sync	Char	Sync character. The ASCII message is always preceded by a single ‘#’ symbol	N
2	Message	Char	The ASCII name of the log or command	N
3	Port	Char	The name of the port from which the log was generated. The string is made up of the port name followed by an _x where x is a number from 1 to 31 denoting the virtual address of the port. If no virtual address is indicated, it is assumed to be address 0	Y
4	Sequence #	Long	Used for multiple related logs. It is a number that counts down from N-1 to 0, where 0 means it is the last one of the set. Most logs only come out one at a time in which case this number is 0	N
5	% Idle Time	Float	The minimum percentage of time the processor is idle, between successive logs, with the same Message ID	Y
6	Time Status	Enum	The value indicates the quality of the GPS reference time (see *Table 10, GPS Reference Time Status* on page 23)	Y
7	Week	Ulong	GPS reference week number	Y
8	Seconds	GPSec	Seconds from the beginning of the GPS reference week; accurate to the millisecond level	Y
9	Receiver Status	Ulong	An eight digit hexadecimal number representing the status of various hardware and software components of the receiver, between successive logs, with the same Message ID (see *Table 3, Receiver Status* on page 13)	Y
10	Reserved	Ulong	Reserved for internal use.	Y
11	Receiver S/W Version	Ulong	A value (0 - 65535) representing the receiver software build number	Y
12	;	Char	The character indicates the end of the header	N

Example Log:

```
#RAWEPHEMA,COM1,0,35.0,SATTIME,1364,496230.000,00100000,97b7,2310;
30,1364,496800,8b0550a1892755100275e6a09382232523a9dc04ee6f794a0000090394ee,8b05
50a189aa6ff925386228f97eabf9c8047e34a70ec5a10e486e794a7a,8b0550a18a2effc2f80061c
2fffc267cd09f1d5034d3537affa28b6ff0eb*7a22f279
```

Table 3: Receiver Status

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Error flag	No error	Error
	1	0x00000002	Temperature status	Within specifications	Warning
	2	0x00000004	Voltage supply status	OK	Warning
	3	0x00000008	Antenna power status	Powered	Not powered
N1	4	0x00000010	LNA Failure		
	5	0x00000020	Antenna open flag [a]	OK	Open
	6	0x00000040	Antenna shorted flag [a]	OK	Shorted
	7	0x00000080	CPU overload flag [a]	No overload	Overload
N2	8	0x00000100	COM1 buffer overrun flag	No overrun	Overrun
	9	0x00000200	COM2 buffer overrun flag	No overrun	Overrun
	10	0x00000400	COM3 buffer overrun flag	No overrun	Overrun
	11	0x00000800	Link overrun flag	No overrun	Overrun
N3	12	0x00001000	Reserved		
	13	0x00002000	Aux transmit overrun flag	No overrun	Overrun
	14	0x00004000	AGC out of range		
	15	0x00008000	Reserved		
N4	16	0x00010000	INS Reset	No Reset	INS filter has reset
	17	0x00020000	Reserved		
	18	0x00040000	Almanac flag/UTC known	Valid	Invalid
	19	0x00080000	Position solution flag	Valid	Invalid
N5	20	0x00100000	Position fixed flag	Not fixed	Fixed
	21	0x00200000	Clock steering status	Enabled	Disabled
	22	0x00400000	Clock model flag	Valid	Invalid
	23	0x00800000	External oscillator locked flag	Unlocked	Locked
N6	24	0x01000000	Software resource	OK	Warning
	25	0x02000000	Reserved		
	26	0x04000000			
	27	0x08000000			
N7	28	0x10000000			
	29	0x20000000	Auxiliary 3 status event flag	No event	Event
	30	0x40000000	Auxiliary 2 status event flag	No event	Event
	31	0x80000000	Auxiliary 1 status event flag	No event	Event

a. This flag is only available on certain products.

2.1.2 Abbreviated ASCII

This message format is designed to make entering and viewing commands and logs simple. The data is represented as simple ASCII characters, separated by spaces or commas and arranged in an easy to understand format. There is also no 32-bit CRC for error detection because it is meant for viewing by the user.

Example Command:

```
log com1 loglist
```

Resultant Log:

```
<LOGLIST COM1 0 69.0 FINE 0 0.000 00240000 206d 0
<     4
<          COM1 RXSTATUSEVENTA ONNEW 0.000000 0.000000 NOHOLD
<          COM2 RXSTATUSEVENTA ONNEW 0.000000 0.000000 NOHOLD
<          COM3 RXSTATUSEVENTA ONNEW 0.000000 0.000000 NOHOLD
<          COM1 LOGLIST ONCE 0.000000 0.000000 NOHOLD
```

The array of 4 logs are offset from the left hand side and start with ‘<’.

2.1.3 Binary

Binary messages are strictly machine readable format. They are ideal for applications where the amount of data transmitted is fairly high. Due to the inherent compactness of binary as opposed to ASCII data, messages are much smaller. The smaller message size allows a larger amount of data to be transmitted and received by the receiver’s communication ports. The structure of all binary messages follows the general conventions as noted here:

1. Basic format of:

   Header 3 Sync bytes plus 25-bytes of header information. The header length is variable as fields may be appended in the future. Always check the header length.

   Data variable

   CRC 4 bytes

2. The 3 Sync bytes will always be:

Byte	Hex	Decimal
First	AA	170
Second	44	68
Third	12	18

3. The CRC is a 32-bit CRC (see *32-Bit CRC* on page 25 for the CRC algorithm) performed on all data including the header.

4. The header is in the format shown in *Table 4, Binary Message Header Structure* on page 15.

Table 4: Binary Message Header Structure

Field	Field Name	Field Type	Description	Binary Bytes	Binary Offset	Ignored on Input
1	Sync	Char	Hexadecimal 0xAA	1	0	N
2	Sync	Char	Hexadecimal 0x44	1	1	N
3	Sync	Char	Hexadecimal 0x12	1	2	N
4	Header Length	Uchar	Length of the header	1	3	N
5	Message ID	Ushort	This is the Message ID number of the log	2	4	N
6	Message Type	Char	Bits 0-4 = Reserved Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response bit (see *Responses* on page 20) 0 = Original Message 1 = Response Message	1	6	N
7	Port Address	Uchar	See *Table 5, Detailed Port Identifier* on page 16 (decimal values >=32 may be used) (lower 8-bits only) [a]	1	7	N [b]
8	Message Length	Ushort	The length in bytes of the body of the message, not including the header nor the CRC	2	8	N
9	Sequence	Ushort	Used for multiple related logs. It is a number that counts down from N-1 to 0 where N is the number of related logs and 0 means it is the last one of the set. Most logs only come out one at a time in which case this number is 0	2	10	N
10	Idle Time	Uchar	Time the processor is idle, in the last second between successive logs, with the same Message ID. Take the time (0 - 200) and divide by two to give the percentage of time (0 - 100%)	1	12	Y
11	Time Status	Enum	Indicates the quality of the GPS reference time (see *Table 10, GPS Reference Time Status* on page 23).	1 [c]	13	N [d]
12	Week	Ushort	GPS reference week number	2	14	N
13	ms	GPSec	Milliseconds from the beginning of the GPS reference week	4	16	N

Field	Field Name	Field Type	Description	Binary Bytes	Binary Offset	Ignored on Input
14	Receiver Status	Ulong	32-bits representing the status of various hardware and software components of the receiver, between successive logs, with the same Message ID (see *Table 3, Receiver Status* on page 13)	4	20	Y
15	Reserved	Ushort	Reserved for internal use	2	24	Y
16	Receiver S/W Version	Ushort	A value (0 - 65535) representing the receiver software build number	2	26	Y

a. The 8-bit size means you will only see 0xA0 to 0xBF when the top bits are dropped from a port value greater than 8-bits. For example, ASCII port USB1 will be seen as 0xA0 in the binary output.

b. Recommended value is THISPORT (binary 192).

c. This ENUM is not 4-bytes long but, as indicated in the table, is only 1-byte.

d. Fields 12 and 13 (Week and ms) are ignored if Field 11 (Time Status) is invalid. In this case, the current receiver time is used. The recommended values for the three time fields are 0, 0, 0.

Table 5: Detailed Port Identifier

ASCII Port Name	Hex Port Value	Decimal Port Value [a]	Description
NO_PORTS	0	0	No ports specified
COM1_ALL	1	1	All virtual ports for COM1
COM2_ALL	2	2	All virtual ports for COM2
COM3_ALL	3	3	All virtual ports for COM3
THISPORT_ALL	6	6	All virtual ports for the current port
ALL_PORTS	8	8	All virtual ports for all ports
XCOM1_ALL	9	9	All virtual ports for XCOM1
XCOM2_ALL	10	10	All virtual ports for XCOM2
USB1_ALL	d	13	All virtual ports for USB1
USB2_ALL	e	14	All virtual ports for USB2
USB3_ALL	f	15	All virtual ports for USB3
AUX_ALL	10	16	All virtual ports for the AUX[b]
XCOM3_ALL	11	17	All virtual XCOM3
ICOM1_ALL	17	23	All virtual ports for ICOM1
ICOM2_ALL	18	24	All virtual ports for ICOM2
ICOM3_ALL	19	25	All virtual ports for ICOM3
NCOM1_ALL	1a	26	All virtual ports for NCOM1
NCOM2_ALL	1b	27	All virtual ports for NCOM2
NCOM3_ALL	1c	28	All virtual ports for NCOM3
COM1	20	32	COM1, virtual port 0

ASCII Port Name	Hex Port Value	Decimal Port Value [a]	Description
COM1_1	21	33	COM1, virtual port 1
. . .			
COM1_31	3f	63	COM1, virtual port 31
COM2	40	64	COM2, virtual port 0
. . .			
COM2_31	5f	95	COM2, virtual port 31
COM3	60	96	COM3, virtual port 0
. . .			
COM3_31	7f	127	COM3, virtual port 31
SPECIAL	a0	160	Unknown port, virtual port 0
. . .			
SPECIAL_31	bf	191	Unknown port, virtual port 31
THISPORT	c0	192	Current COM port, virtual port 0
. . .			
THISPORT_31	df	223	Current COM port, virtual port 31
XCOM1	1a0	416	XCOM1, virtual port 0
XCOM1_1	1a1	417	XCOM1, virtual port 1
. . .			
XCOM1_31	1bf	447	XCOM1, virtual port 31
XCOM2	2a0	672	XCOM2, virtual port 0
XCOM2_1	2a1	673	XCOM2, virtual port 1
. . .			
XCOM2_31	2bf	703	XCOM2, virtual port 31
USB1	5a0	1440	USB1, virtual port 0
USB1_1	5a1	1441	USB1, virtual port 1
. . .			
USB1_31	5bf	1471	USB1, virtual port 31
USB2	6a0	1696	USB2, virtual port 0
. . .			
USB2_31	6bf	1727	USB2, virtual port 31
USB3	7a0	1952	USB3, virtual port 0
. . .			
USB3_31	7bf	1983	USB port 3, virtual port 31
AUX	8a0	2208	AUX port, virtual port 0 [b]
. . .			

ASCII Port Name	Hex Port Value	Decimal Port Value [a]	Description
AUX_31	8bf	2239	AUX port, virtual port 31 [b]
XCOM3	9a0	2464	XCOM3, virtual port 0
. . .			
XCOM3_31	9bf	2495	XCOM3, virtual port 31
ICOM1	fa0	4000	ICOM1, virtual port 0
. . .			
ICOM1_31	fbf	4031	ICOM1, virtual port 31
ICOM2	10a0	4256	ICOM2, virtual port 0
. . .			
ICOM2_31	10bf	4287	ICOM2, virtual port 31
ICOM3	11a0	4512	ICOM3, virtual port 0
. . .			
ICOM3_31	11bf	4543	ICOM3, virtual port 31
NCOM1	12a0	4768	NCOM1, virtual port 0
. . .			
NCOM1_31	12bf	4799	NCOM1, virtual port 31
NCOM2	13a0	5024	NCOM2, virtual port 0
. . .			
NCOM2_31	13bf	5055	NCOM2, virtual port 31
NCOM3	14a0	5280	NCOM3, virtual port 0
. . .			
NCOM3_31	14bf	5311	NCOM3, virtual port 31

a. Decimal port values 0 through 16 are only available to the UNLOGALL command and cannot be used in the UNLOG command or in the binary message header (see *Table 4, Binary Message Header Structure* on page 15).

b. The AUX port is only available on specific products.

> COM1_ALL, COM2_ALL, COM3_ALL, THISPORT_ALL, ALL_PORTS, USB1_ALL, USB2_ALL, USB3_ALL, AUX_ALL, ICOM1_ALL, ICOM2_ALL, ICOM3_ALL, NCOM1_ALL, NCOM2_ALL, XCOM1_ALL, XCOM2_ALL, XCOM3_ALL and NCOM3_ALL are only valid for the UNLOGALL command.

2.1.4 Description of ASCII and Binary Logs with Short Headers

These logs are set up in the same way as normal ASCII or binary logs except a normal ASCII or binary header is replaced with a short header (see *Table 6, Short ASCII Message Header Structure* on page 19 and *Table 7, Short Binary Message Header Structure* on page 19).

Table 6: Short ASCII Message Header Structure

Field	Field Name	Field Type	Description
1	%	Char	% symbol
2	Message	Char	This is the name of the log
3	Week Number	Ushort	GNSS week number
4	Milliseconds	Ulong	Milliseconds from the beginning of the GNSS week

Table 7: Short Binary Message Header Structure

Field	Field Name	Field Type	Description	Binary Bytes	Binary Offset
1	Synch	Char	Hex 0xAA	1	0
2	Synch	Char	Hex 0x44	1	1
3	Synch	Char	Hex 0x13	1	2
4	Message Length	Uchar	Message length, not including header or CRC	1	3
5	Message ID	Ushort	Message ID number	2	4
6	Week Number	Ushort	GNSS week number	2	6
7	Milliseconds	Ulong	Milliseconds from the beginning of the GNSS week	4	8

2.2 Responses

By default, if you input a message you get back a response. If desired, the INTERFACEMODE command can be used to disable response messages (see *INTERFACEMODE* on page 60). The response will be in the exact format you entered the message (that is, binary input = binary response).

2.2.1 Abbreviated Response

Just the leading '<' followed by the response string, for example: <OK.

2.2.2 ASCII Response

Full header with the message name being identical except ending in an 'R' (for response). The body of the message consists of a 40 character string for the response string.
For example:

#BESTPOSR,COM1,0,67.0,FINE,1028,422060.400,00000000,a31b,0;"OK" *b867caad

2.2.3 Binary Response

Similar to an ASCII response except that it follows the binary protocols, see *Table 8, Binary Message Response Structure* on page 21.

Table 9, Binary Message Sequence on page 22 is an example of the sequence for requesting and then receiving BESTPOSB. The example is in hex format. When you enter a hex command, you may need to add a ‘\x’ or ‘0x’ before each hex pair, depending on your code.
For example:

0xAA0x440x120x1C0x010x000x02 (and so on).

Table 8: Binary Message Response Structure

	Field	Field Name	Field Type	Description	Binary Bytes	Binary Offset
BINARY HEADER	1	Sync	Char	Hexadecimal 0xAA	1	0
	2	Sync	Char	Hexadecimal 0x44	1	1
	3	Sync	Char	Hexadecimal 0x12	1	2
	4	Header Lgth	Uchar	Length of the header	1	3
	5	Message ID	Ushort	Message ID number	2	4
	6	Message Type	Char	Bit 7 = Response Bit 1 = Response Message	1	6
	7	Port Address	Uchar	See *Table 5, Detailed Port Identifier* on page 16	1	7
	8	Message Length	Ushort	The length in bytes of the body of the message (not the CRC)	2	8
	9	Sequence	Ushort	Normally 0	2	10
	10	Idle Time	Uchar	Idle time	1	12
	11	Time Status	Enum	*Table 10, GPS Reference Time Status* on page 23	1 [a]	13
	12	Week	Ushort	GPS reference week number	2	14
	13	ms	GPSec	Milliseconds into GPS reference week	4	16
	14	Receiver Status	Ulong	*Table 3, Receiver Status* on page 13	4	20
	15	Reserved	Ushort	Reserved for internal use	2	24
	16	Receiver S/W Version	Ushort	Receiver software build number	2	26
ID	17	Response ID	Enum	The output format of the messages is dependent on the format of the input command. If the command is input as abbreviated ASCII, the output will be abbreviated ASCII. The same rule applies for both ASCII and binary formats (*Table 11, Response Messages* on page 27)	4	28
HEX	18	Response	Hex	String containing the ASCII response in hex coding to match the ID above (for example, 0x4F4B = OK)	variable	32

a. This ENUM is not 4-bytes long but as indicated in the table is only 1 byte.

Table 9: Binary Message Sequence

Direction	Sequence	Data
To Receiver	LOG Command Header	AA44121C 01000240 20000000 1D1D0000 29160000 00004C00 55525A80
	LOG Parameters	20000000 2A000000 02000000 00000000 0000F03F 00000000 00000000 00000000
	Checksum	2304B3F1
From Receiver	LOG Response Header	AA44121C 01008220 06000000 FFB4EE04 605A0513 00004C00 FFFF5A80
	Log Response Data	01000000 4F4B
	Checksum	DA8688EC
From Receiver	BESTPOSB Header	AA44121C 2A000220 48000000 90B49305 B0ABB912 00000000 4561BC0A
	BESTPOSB Data	00000000 10000000 1B0450B3 F28E4940 16FA6BBE 7C825CC0 0060769F 449F9040 A62A82C1 3D000000 125ACB3F CD9E983F DB664040 00303030 00000000 00000000 0B0B0000 00060003
	Checksum	42DC4C48

2.3 GLONASS Slot and Frequency Numbers

When a PRN in a log is in the range 38 to 61, then that PRN represents a GLONASS Slot where the Slot shown is the actual GLONASS Slot Number plus 37.

Similarly, the GLONASS Frequency shown in logs is the actual GLONASS Frequency plus 7.

For example:

```
#SATVISA,COM1,0,53.5,FINESTEERING,1363,234894.000,00000000,0947,2277;
TRUE,TRUE,46,
2,0,0,73.3,159.8,934.926,934.770,
...
43,8,0,-0.4,163.7,4528.085,4527.929,
...
3,0,0,-79.9,264.3,716.934,716.778*b94813d3
```

where 2 and 3 are GPS satellites and 43 is a GLONASS satellite. It's actual GLONASS Slot Number is 6. The SATVIS log shows 43 (6+37). It's actual GLONASS frequency is 1. The SATVIS log shows 8 (1+7).

Refer to *An Introduction to GNSS*, available on our website at www.novatel.com/an-introduction-to-gnss/ for more information.

2.4 GPS Reference Time Status

All reported receiver times are subject to a qualifying time status. The status indicates how well a time is known (see *Table 10, GPS Reference Time Status* on page 23).

Table 10: GPS Reference Time Status

GPS Reference Time Status (Decimal)	GPS Reference Time Status[a] (ASCII)	Description
20	UNKNOWN	Time validity is unknown
60	APPROXIMATE	Time is set approximately
80	COARSEADJUSTING	Time is approaching coarse precision
100	COARSE	This time is valid to coarse precision
120	COARSESTEERING	Time is coarse set and is being steered
130	FREEWHEELING	Position is lost and the range bias cannot be calculated
140	FINEADJUSTING	Time is adjusting to fine precision
160	FINE	Time has fine precision
170	FINEBACKUPSTEERING	Time is fine set and is being steered by the backup system
180	FINESTEERING	Time is fine set and is being steered
200	SATTIME	Time from satellite. Only used in logs containing satellite data such as ephemeris and almanac

a. See also *Message Time Stamps* on page 24.

There are several distinct states the receiver goes through when CLOCKADJUST is enabled:

- UNKNOWN (initial state)
- COARSESTEERING (initial coarse time set)
- FINESTEERING (normal operating state)
- FINEBACKUPSTEERING (when the backup system is used for time)
- FREEWHEELING (when range bias becomes unknown)

and when the CLOCKADJUST is disabled:

- UNKNOWN (initial state)
- COARSE (initial coarse time set)
- FINE (normal operating state)

On start up and before any satellites are being tracked, the receiver can not possibly know the current time. As such, the receiver time starts counting at GPS reference week 0 and second 0.0. The time status flag is set to UNKNOWN.

If time is input to the receiver using the SETAPPROXTIME command or on receipt of an RTCAEPHEM message the time status will be APPROXIMATE.

After the first ephemeris is decoded, the receiver time is set to a resolution of ±10 milliseconds. This state is qualified by the COARSE or COARSESTEERING time status flag depending on the state of the CLOCKADJUST switch.

Once a position is known and range biases are being calculated, the internal clock model will begin modeling the position range biases and the receiver clock offset.

Modelling will continue until the model is a good estimation of the actual receiver clock behavior. At this time, the receiver time will again be adjusted, this time to an accuracy of ±1 microsecond. This state is qualified by the FINE time status flag.

The final logical time status flag depends on whether CLOCKADJUST is enabled or not. If CLOCKADJUST is disabled, the time status flag will never improve on FINE. The time will only be adjusted again to within ±1 microsecond if the range bias gets larger than ±250 milliseconds. If CLOCKADJUST is enabled, the time status flag is set to FINESTEERING and the receiver time is continuously updated (steered) to minimize the receiver range bias.

When the backup system is used, the time status is set to FINEBACKUPSTEERING. If, for some reason, the position is lost and the range bias cannot be calculated, the time status is degraded to FREEWHEELING.

2.5 Message Time Stamps

All NovAtel format messages generated by the OEM6 family of receivers have a GPS reference time stamp in their header. GPS reference time is referenced to UTC with zero point defined as midnight on the night of January 5, 1980. The time stamp consists of the number of weeks since that zero point and the number of seconds since the last week number change (0 to 604,799). GPS reference time differs from UTC time since leap seconds are occasionally inserted into UTC and GPS reference time is continuous. In addition, a small error (less than 1 microsecond) can exist in synchronization between UTC and GPS reference time. The TIME log reports both GNSS and UTC time and the offset between the two.

The data in synchronous logs (for example, RANGE, BESTPOS, TIME) are based on a periodic measurement of satellite pseudoranges. The time stamp on these logs is the receiver estimate of GPS reference time at the time of the measurement. When setting time in external equipment, a small synchronous log with a high baud rate will be accurate to a fraction of a second. A synchronous log with trigger ONTIME 1 can be used in conjunction with the 1PPS signal to provide relative accuracy better than 250 ns.

Other log types (asynchronous and polled) are triggered by an external event and the time in the header may not be synchronized to the current GPS reference time. Logs that contain satellite broadcast data (for example, ALMANAC, GPSEPHEM) have the transmit time of their last subframe in the header. In the header of differential time matched logs (for example, MATCHEDPOS) is the time of the matched reference and local observation that they are based on. Logs triggered by a mark event (for example, MARKEDPOS, MARKTIME) have the estimated GPS reference time of the mark event in their header. In the header of polled logs (for example, LOGLIST, PORTSTATS, VERSION) is the approximate GPS reference time when their data was generated. However, when asynchronous logs are triggered ONTIME, the time stamp will represent the time the log was generated and not the time given in the data.

2.6 Decoding of the GPS Reference Week Number

The GPS reference week number provided in the raw satellite data is the 10 least significant bits (or 8 least significant bits in the case of the almanac data) of the full week number. When the receiver processes the satellite data, the week number is decoded in the context of the current era and therefore is computed as the full week number starting from week 0 or January 6, 1980. Therefore, in all log headers and decoded week number fields, the full week number is given. Only in raw data, such as the *data* field of the RAWALM log or the *subframe* field of the RAWEPHEM log, will the week number remain as the 10 (or 8) least significant bits.

2.7 32-Bit CRC

The ASCII and Binary OEM6 family message formats all contain a 32-bit CRC for data verification. This allows the user to ensure the data received (or transmitted) is valid with a high level of certainty. This CRC can be generated using the following C algorithm:

```
#define CRC32_POLYNOMIAL   0xEDB88320L
/* --------------------------------------------------------------------------
Calculate a CRC value to be used by CRC calculation functions.
-------------------------------------------------------------------------- */
unsigned long CRC32Value(int i)
{
   int j;
   unsigned long ulCRC;
   ulCRC = i;
   for ( j = 8 ; j > 0; j-- )
   {
      if ( ulCRC & 1 )
         ulCRC = ( ulCRC >> 1 ) ^ CRC32_POLYNOMIAL;
      else
         ulCRC >>= 1;
   }
   return ulCRC;
}
/* --------------------------------------------------------------------------
Calculates the CRC-32 of a block of data all at once
-------------------------------------------------------------------------- */
unsigned long CalculateBlockCRC32(
   unsigned long ulCount,     /* Number of bytes in the data block */
   unsigned char *ucBuffer ) /* Data block */
{
   unsigned long ulTemp1;
   unsigned long ulTemp2;
   unsigned long ulCRC = 0;
  while ( ulCount-- != 0 )
   {
      ulTemp1 = ( ulCRC >> 8 ) & 0x00FFFFFFL;
      ulTemp2 = CRC32Value( ((int) ulCRC ^ *ucBuffer++ ) & 0xff );
      ulCRC = ulTemp1 ^ ulTemp2;
   }
   return( ulCRC );
}
```

The NMEA checksum is an XOR of all the bytes (including delimiters such as ',' but excluding the * and $) in the message output. It is therefore an 8-bit and not a 32-bit checksum.

Not all logs may be available. Every effort is made to ensure examples are correct, however, a checksum may be created for promptness in publication. In this case it will appear as ‘9999’.

Example:

BESTPOSA and BESTPOSB from an OEM6 family receiver.

ASCII:

```
#BESTPOSA,COM1,0,78.0,FINESTEERING,1427,325298.000,00000000,6145,2748;
SOL_COMPUTED,SINGLE,51.11678928753,-114.03886216575,1064.3470,-16.2708,
WGS84,2.3434,1.3043,4.7300,"",0.000,0.000,7,7,0,0,0,06,0,03*9c9a92bb
```

BINARY:

```
0xAA, 0x44, 0x12, 0x1C 0x2A, 0x00, 0x02, 0x20, 0x48, 0x00, 0x00, 0x00, 0x90, 0xB4,
0x93, 0x05, 0xB0, 0xAB, 0xB9, 0x12, 0x00, 0x00, 0x00, 0x00, 0x45, 0x61, 0xBC, 0x0A,
0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x1B, 0x04, 0x50, 0xB3, 0xF2, 0x8E,
0x49, 0x40, 0x16, 0xFA, 0x6B, 0xBE, 0x7C, 0x82, 0x5C, 0xC0, 0x00, 0x60, 0x76, 0x9F,
0x44, 0x9F, 0x90, 0x40, 0xA6, 0x2A, 0x82, 0xC1, 0x3D, 0x00, 0x00, 0x00, 0x12, 0x5A,
0xCB, 0x3F, 0xCD, 0x9E, 0x98, 0x3F, 0xDB, 0x66, 0x40, 0x40, 0x00, 0x30, 0x30, 0x30,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B, 0x0B, 0x00, 0x00, 0x00, 0x06,
0x00, 0x03,
0x42, 0xdc, 0x4c, 0x48
```

Below is a demonstration of how to generate the CRC from both ASCII and BINARY messages using the function described above.

> When you pass the data into the code that follows, exclude the checksum shown in ***bold italics*** above.

ASCII:

```
#include <iostream.h>
#include <string.h>
void  main()
{
char *i = ”BESTPOSA,COM2,0,77.5,FINESTEERING,1285,160578.000,00000020,5941,1164;
SOL_COMPUTED,SINGLE,51.11640941570,-114.03830951024,1062.6963,-16.2712,
WGS84,1.6890,1.2564,2.7826,\"\",0.000,0.000,10,10,0,0,0,0,0,0";
unsigned long iLen = strlen(i);
unsigned long CRC = CalculateBlockCRC32(iLen, (unsigned char*)i);
cout << hex << CRC <<endl;
}
```

BINARY:

```
#include <iostream.h>
#include <string.h>
int main()
{
unsigned char buffer[] = {0xAA, 0x44, 0x12, 0x1C, 0x2A, 0x00, 0x02, 0x20, 0x48,
0x00, 0x00, 0x00, 0x90, 0xB4, 0x93, 0x05, 0xB0, 0xAB, 0xB9, 0x12, 0x00, 0x00, 0x00,
0x00, 0x45, 0x61, 0xBC, 0x0A, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x1B,
0x04, 0x50, 0xB3, 0xF2, 0x8E, 0x49, 0x40, 0x16, 0xFA, 0x6B, 0xBE, 0x7C, 0x82, 0x5C,
0xC0, 0x00, 0x60, 0x76, 0x9F, 0x44, 0x9F, 0x90, 0x40, 0xA6, 0x2A, 0x82, 0xC1, 0x3D,
0x00, 0x00, 0x00, 0x12, 0x5A, 0xCB, 0x3F, 0xCD, 0x9E, 0x98, 0x3F, 0xDB, 0x66, 0x40,
0x40, 0x00, 0x30, 0x30, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0B,
0x0B, 0x00, 0x00, 0x00, 0x06, 0x00, 0x03};
unsigned long crc = CalculateBlockCRC32(sizeof(buffer)buffer);
cout << hex << crc <<endl;
//Please note that this hex needs to be reversed due to Big Endian order where the
most significant value in the sequence is stored first (at the lowest storage
address). For example, the two bytes required for the hex number 4F52 is stored as
524F.
}
```

Chapter 3 Responses

The receiver is capable of outputting several responses for various conditions. Most responses are error messages to indicate when something is not correct.

The output format of the messages is dependent on the format of the input command. If the command is input as abbreviated ASCII, the output will be abbreviated ASCII. The same rule applies for both ASCII and binary formats.

Table 11: Response Messages

ASCII Message	Binary Message ID	Meaning
OK	1	Command was received correctly
REQUESTED LOG DOES NOT EXIST	2	The log requested does not exist
NOT ENOUGH RESOURCES IN SYSTEM	3	The request has exceeded a limit (for example, the maximum number of logs are being generated)
DATA PACKET DOESN'T VERIFY	4	Data packet is not verified
COMMAND FAILED ON RECEIVER	5	Command did not succeed in accomplishing requested task
INVALID MESSAGE ID	6	The input message ID is not valid
INVALID MESSAGE. FIELD = X	7	Field x of the input message is not correct
INVALID CHECKSUM	8	The checksum of the input message is not correct. Only applies to ASCII and binary format messages.
MESSAGE MISSING FIELD	9	A field is missing from the input message
ARRAY SIZE FOR FIELD X EXCEEDS MAX	10	Field x contains more array elements than allowed
PARAMETER X IS OUT OF RANGE	11	Field x of the input message is outside the acceptable limits
TRIGGER X NOT VALID FOR THIS LOG	14	Trigger type x is not valid for this type of log
AUTHCODE TABLE FULL - RELOAD SOFTWARE	15	Too many authcodes are stored in the receiver. The receiver firmware must be reloaded
INVALID DATE FORMAT	16	This error is related to the inputting of authcodes. Indicates the date attached to the code is not valid
INVALID AUTHCODE ENTERED	17	The authcode entered is not valid
NO MATCHING MODEL TO REMOVE	18	The model requested for removal does not exist
NOT VALID AUTH CODE FOR THAT MODEL	19	The model attached to the authcode is not valid
CHANNEL IS INVALID	20	The selected channel is invalid
REQUESTED RATE IS INVALID	21	The requested rate is invalid

ASCII Message	Binary Message ID	Meaning
WORD HAS NO MASK FOR THIS TYPE	22	The word has no mask for this type of log
CHANNELS LOCKED DUE TO ERROR	23	Channels are locked due to error
INJECTED TIME INVALID	24	Injected time is invalid
COM PORT NOT SUPPORTED	25	The COM or USB port is not supported
MESSAGE IS INCORRECT	26	The message is invalid
INVALID PRN	27	The PRN is invalid
PRN NOT LOCKED OUT	28	The PRN is not locked out
PRN LOCKOUT LIST IS FULL	29	PRN lockout list is full
PRN ALREADY LOCKED OUT	30	The PRN is already locked out
MESSAGE TIMED OUT	31	Message timed out
UNKNOWN COM PORT REQUESTED	33	Unknown COM or USB port requested
HEX STRING NOT FORMATTED CORRECTLY	34	Hex string not formatted correctly
INVALID BAUD RATE	35	The baud rate is invalid
MESSAGE IS INVALID FOR THIS MODEL	36	Message is invalid for this model of receiver
COMMAND ONLY VALID IF IN NVM FAIL MODE	40	Command is only valid if NVM is in fail mode
INVALID OFFSET	41	The offset is invalid
MAXIMUM NUMBER OF USER MESSAGES REACHED	78	Maximum number of user messages has been reached
GPS PRECISE TIME IS ALREADY KNOWN	84	GPS precise time is already known
ZUPT DISABLED BY USER	149	An INSZUPT command was sent after INSZUPTCONTROL command was used to disable the use of ZUPTs.
IMU SPECS LOCKED FOR THIS IMU TYPE	150	SPAN allows the default specifications for a select few IMUs to be modified to support different variants. However, most IMU specifications are not allowed to change.
COMMAND INVALID FOR THIS IMU	154	The entered command cannot be used with the configured IMU. For example, the LEVERARMCALIBRATE command is not valid for lower quality IMUs.

ASCII Message	Binary Message ID	Meaning
IMU TYPE IS NOT SUPPORTED WITH CURRENT MODEL	157	A firmware model upgrade is required to use the requested IMU (CONNECTIMU command).
TRIGGER BUFFER IS FULL	161	The TIMEDEVENTPULSE limit of 10 events has been reached, and a new event cannot be set until an event is cleared.
SETUPSENSOR COMMAND IS LOCKED	163	The SETUPSENSOR command cannot be modified because there are remaining trigger events queued.

Chapter 4 SPAN Commands

The SPAN specific commands are described in this chapter.

For information about other available commands, refer to the OEM6 Family Firmware Reference Manual (OM-20000129).

4.1 Using a Command as a Log

All NovAtel commands may be used for data input or used to request data output. INS specific commands can be in Abbreviated ASCII, ASCII or Binary format.

Consider the `lockout` command with the syntax:

```
lockout prn
```

You can put this command into the receiver to de-weight an undesirable satellite in the solution, or use the `lockout` command as a log to see if there is a satellite PRN already locked out. In ASCII, this might be:

```
log com1 lockouta once
```

Notice the 'a' after `lockout` to signify you are looking for ASCII output.

> Ensure all windows, other than the Console, are closed in NovAtel Connect and then use the SAVECONFIG command to save settings in NVM. Otherwise, unnecessary data logging occurs and may overload the system.

4.2 SPAN Specific Commands

Table 12: SPAN Commands for OEM6 - Alphabetical

ASCII Name	Description	Message ID
ALIGNMENTMODE	Sets the Alignment Mode	1214
APPLYVEHICLEBODYROTATION	Enables or disables vehicle body rotation	1071
ASYNCHINSLOGGING	Enables or disables the asynchronous INS logs	1363
CONNECTIMU	Connects an IMU to a port	1428
DUALANTENNAPORTCONFIG	Select dual antenna source port	1356
ENCLOSURECOMSELECT	Sets the external COM port to COM3 or USB	1536
ENCLOSUREWHEELSENSOR	Control Wheel Sensor Data	1535
EVENTINCONTROL	Control Event-In properties	1637
EVENTOUTCONTROL	Control Event-Out properties	1636
EXTERNALPVAS	External PVA update	1463
EXTHDGOFFSET	Set the angular offset	1204
GIMBALSPANROTATION	Gimbal frame to SPAN frame rotation	1319
HEAVEFILTER	Enables or disables heave filtering	1427

ASCII Name	Description	Message ID
IMUCONFIGURATION	Send one-time configuration data to an IMU	1745
INPUTGIMBALANGLE	Enters gimbal angles into the receiver	1317
INSCOMMAND	Enables or disables INS positioning	379
INSTHRESHOLDS	Change the INS_HIGH_VARIANCE Threshold	1448
INSWHEELUPDATE	Enables or Disables Use of a Wheel Sensor	647
INSZUPT	Requests a Zero Velocity Update (ZUPT)	382
INSZUPTCONTROL	Enables or disables Zero Velocity Updates (ZUPT)	1293
INTERFACEMODE	Sets interface type for a port	3
LEVERARMCALIBRATE	Controls the IMU to antenna lever arm calibration	675
NMEATALKER	Sets the NMEA Talker ID	861
RELINSAUTOMATION	Enables or disables Relative INS on the rover receiver	1763
RELINSCONFIG	Configure Relative INS	1797
RVBCALIBRATE	Enables or disables the calculation of the vehicle to SPAN body angular offset	641
SETALIGNMENTVEL	Sets the minimum required velocity for a kinematic alignment	1397
SETGIMBALORIENTATION	Converts the Mount Body frame to the Mount Computation frame for SPAN	1318
SETHEAVEWINDOW	Sets the length of the heave filter	1383
SETIMUORIENTATION	Sets the IMU axis that is aligned with gravity	567
SETIMUPORTPROTOCOL	Sets the protocol used for the IMU serial port	1767
SETIMUSPECS	Sets the error specifications and data rate for the IMU	1295
SETIMUTOANTOFFSET	Sets the IMU to antenna offset for the primary antenna	383
SETIMUTOANTOFFSET2	Sets the IMU to antenna offset for the secondary antenna	1205
SETIMUTOGIMBALOFFSET	Sets the IMU to gimbal mount offset	1352
SETINITATTITUDE	Sets the initial attitude of the SPAN system in degrees	862
SETINITAZIMUTH	Sets the initial azimuth and standard deviation	863
SETINSOFFSET	Sets an offset from the IMU for the output position and velocity of the INS solution	676
SETINSROTATION	Set the Rotation of the Attitude Output	1796
SETINSUPDATE	Enable/Disable INS Filter Updates	1821
SETMARK1OFFSET	Sets the offset to the Mark1 trigger event	1069
SETMARK2OFFSET	Sets the offset to the Mark2 trigger event	1070
SETMARK3OFFSET	Sets the offset to the Mark3 trigger event	1116

ASCII Name	Description	Message ID
SETMARK4OFFSET	Sets the offset to the Mark4 trigger event	1117
SETMAXALIGNMENTTIME	Set a Time Limit for Static Course Alignment	1800
SETRELINSOUTPUTFRAME	Sets the Relative INS Output Frame	1775
SETUPSENSOR	Add a new sensor object	1333
SETWHEELPARAMETERS	Sets the wheel parameters	847
SETWHEELSOURCE	Sets the wheel sensor input source	1722
TAGNEXTMARK	Tags the next incoming mark event	1257
TIMEDEVENTPULSE	Add a new camera event	1337
VEHICLEBODYROTATION	Set the angular offsets between the vehicle frame and the SPAN body frame	642
WHEELVELOCITY	Wheel velocity for INS augmentation	504

Table 13: SPAN Commands for OEM6 - by Message ID

Message ID	ASCII Name	Description
3	INTERFACEMODE	Sets interface type for a port
379	INSCOMMAND	Enables or disables INS positioning
382	INSZUPT	Requests a Zero Velocity Update (ZUPT)
383	SETIMUTOANTOFFSET	Sets the IMU to antenna offset for the primary antenna
504	WHEELVELOCITY	Wheel velocity for INS augmentation
567	SETIMUORIENTATION	Sets the IMU axis that is aligned with gravity
641	RVBCALIBRATE	Enables or disables the calculation of the vehicle to SPAN body angular offset
642	VEHICLEBODYROTATION	Set the angular offsets between the vehicle frame and the SPAN body frame
647	INSWHEELUPDATE	Enables or Disables Use of a Wheel Sensor
675	LEVERARMCALIBRATE	Controls the IMU to antenna lever arm calibration
676	SETINSOFFSET	Sets an offset from the IMU for the output position and velocity of the INS solution
847	SETWHEELPARAMETERS	Sets the wheel parameters
861	NMEATALKER	Sets the NMEA Talker ID
862	SETINITATTITUDE	Sets the initial attitude of the SPAN system in degrees
863	SETINITAZIMUTH	Sets the initial azimuth and standard deviation
1069	SETMARK1OFFSET	Sets the offset to the Mark1 trigger event

Message ID	ASCII Name	Description
1070	SETMARK2OFFSET	Sets the offset to the Mark2 trigger event
1071	APPLYVEHICLEBODYROTATION	Enables or disables vehicle body rotation
1116	SETMARK3OFFSET	Sets the offset to the Mark3 trigger event
1117	SETMARK4OFFSET	Sets the offset to the Mark4 trigger event
1204	EXTHDGOFFSET	Set the angular offset
1205	SETIMUTOANTOFFSET2	Sets the IMU to antenna offset for the secondary antenna
1214	ALIGNMENTMODE	Sets the Alignment Mode
1257	TAGNEXTMARK	Tags the next incoming mark event
1293	INSZUPTCONTROL	Enables or disables Zero Velocity Updates (ZUPT)
1295	SETIMUSPECS	Sets the error specifications and data rate for the IMU
1317	INPUTGIMBALANGLE	Enters gimbal angles into the receiver
1318	SETGIMBALORIENTATION	Converts the Mount Body frame to the Mount Computation frame for SPAN
1319	GIMBALSPANROTATION	Gimbal frame to SPAN frame rotation
1333	SETUPSENSOR	Add a new sensor object
1337	TIMEDEVENTPULSE	Add a new camera event
1352	SETIMUTOGIMBALOFFSET	Sets the IMU to gimbal mount offset
1356	DUALANTENNAPORTCONFIG	Select dual antenna source port
1363	ASYNCHINSLOGGING	Enables or disables the asynchronous INS logs
1383	SETHEAVEWINDOW	Sets the length of the heave filter
1397	SETALIGNMENTVEL	Sets the minimum required velocity for a kinematic alignment
1427	HEAVEFILTER	Enables or disables heave filtering
1428	CONNECTIMU	Connects an IMU to a port
1448	INSTHRESHOLDS	Change the INS_HIGH_VARIANCE Threshold
1463	EXTERNALPVAS	External PVA Update
1535	ENCLOSUREWHEELSENSOR	Control Wheel Sensor Data
1536	ENCLOSURECOMSELECT	Sets the External COM Port to COM3 or USB
1636	EVENTOUTCONTROL	Control Event-Out properties
1637	EVENTINCONTROL	Control Event-In properties
1722	SETWHEELSOURCE	Sets the wheel sensor input source
1745	IMUCONFIGURATION	Send one-time configuration data to an IMU
1763	RELINSAUTOMATION	Enables or disables Relative INS on the rover receiver

Message ID	ASCII Name	Description
1767	SETIMUPORTPROTOCOL	Sets the protocol used for the IMU serial port
1775	SETRELINSOUTPUTFRAME	Sets the Relative INS Output Frame
1796	SETINSROTATION	Set the Rotation of the Attitude Output
1797	RELINSCONFIG	Configure Relative INS
1800	SETMAXALIGNMENTTIME	Set a Time Limit for Static Course Alignment
1821	SETINSUPDATE	Enable/Disable INS Filter Updates

4.2.1 ALIGNMENTMODE
Set the Alignment Mode

Use this command to set the alignment method used by the SPAN system.

The default ALIGNMENTMODE is AUTOMATIC. In this mode, the first available method to align is used. If the receiver is in single antenna operation only the UNAIDED option is available.

Sending the ALIGNMENTMODE command manually overrides the AUTOMATIC setting and allows a specific method to be used.

Message ID: 1214

Abbreviated ASCII Syntax:

```
ALIGNMENTMODE mode
```

Abbreviated ASCII Example:

```
ALIGNMENTMODE AIDED_TRANSFER
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	mode	UNAIDED	0	Regular SPAN static coarse or kinematic alignment mode.	Enum	4	H
		AIDED_STATIC	1	Seed the static coarse alignment with an initial azimuth.			
		AIDED_TRANSFER	2	Seed the full attitude from an ALIGN solution. Pitch and Heading taken from ALIGN, Roll is assumed 0.			
		AUTOMATIC (Default)	3	Seed the full attitude from ALIGN or perform a regular coarse or kinematic alignment, whichever is possible first.			

If the ALIGNMENTMODE selected can use a kinematic alignment (UNAIDED or AUTOMATIC), the `SETIMUORIENTATION` and `VEHICLEBODYROTATION` commands must be sent to the receiver regardless of system configuration and IMU orientation.

4.2.2 APPLYVEHICLEBODYROTATION Enable Vehicle to Body Rotation

Use this command to apply the vehicle to body rotation to the output attitude (that was entered using the `VEHICLEBODYROTATION` command, see page 102). This rotates the SPAN body frame output in the INSPVA, INSPVAS, INSPVAX, INSATT, INSATTS and INSATTX logs to the vehicle frame. APPLYVEHICLEBODYROTATION is disabled by default.

> The `VEHICLEBODYROTATION` command will always be used for Kinematic Alignment procedures regardless of the `APPLYVEHICLEBODYROTATION` state.

Message ID: 1071

Abbreviated ASCII Syntax:

```
APPLYVEHICLEBODYROTATION switch
```

Input Example:

```
APPLYVEHICLEBODYROTATION ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	switch	DISABLE	0	Enable/disable vehicle to body rotation using values entered in the `VEHICLEBODYROTATION` command. The default value is DISABLE	Enum	4	H
		ENABLE	1				

4.2.3 ASYNCHINSLOGGING Enable Asynchronous INS Logs

Use this command to enable or disable the asynchronous INS logs (IMURATECORRIMUS and IMURATEPVAS).

> The asynchronous INS logs are highly advanced logs for users of SPAN on OEM6. The rate controls that limit the output of logs are not applicable to these logs, allowing the user to drive the idle time to zero.
>
> Users of the IMURATECORRIMUS or IMURATEPVAS logs should be limited to those who must have full rate INS solution data, or full rate corrected IMU data, without possible shifts in log time that are present in the synchronous version of these logs.
>
> The asynchronous INS logs are only available at the full rate of the IMU.

Message ID: 1363

Abbreviated ASCII Syntax:

```
ASYNCHINSLOGGING switch
```

Abbreviated ASCII Example:

```
ASYNCHINSLOGGING ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Switch	DISABLE	0	Enable or disable the asynchronous INS logs. The default value is DISABLE.	Enum	4	H
		ENABLE	1				

4.2.4 CONNECTIMU
Connects an IMU to a Port

Use this command to specify the type of IMU connected to the receiver and the receiver port used by the IMU.

Do not use this command for SPAN-CPT or SPAN-IGM receivers.

These receivers have the correct IMU type and IMU port configured by default. Using this command to change these settings will cause the receiver to lose SPAN functionality.

If the IMU port or IMU type are inadvertently changed on a SPAN-CPT or SPAN-IGM, use the FRESET command to change the values back to the default values.

Message ID: 1428

Abbreviated ASCII Syntax:

```
CONNECTIMU IMUPort IMUType
```

Abbreviated ASCII Example:

```
CONNECTIMU COM2 IMU_LN200
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	IMUPort[a]	COM1	1	IMU Port is COM port 1	Enum	4	H
		COM2	2	IMU Port is COM port 2			
		COM3	3	IMU Port is COM port 3			
		COM6	32	IMU Port is COM port 6			
3	IMUType	See *Table 14, IMU Type* on page 39		IMU Type	Enum	4	H+4

a. The IMU-ISA-100, IMU-ISA-100C, IMU-FSAS, IMU-HG1900, IMU-LN200, IMU-µIMU, IMU-CPT and IMU-KVH1750 use RS-422 protocol and must be connected to a receiver port that is configured to use RS-422. Refer to the OEM6 Family Installation and Operation User Manual (OM-20000129) for information about which receiver ports support RS-422 and instructions for enabling RS-422.

For systems with a ProPak6 receiver, the COM3/IMU port must be used.

For systems with an OEM638 receiver, the COM6 port must be used.

Table 14: IMU Type

Binary	ASCII	Description
0	IMU_UNKNOWN	Unknown IMU type (default)
1	IMU_HG1700_AG11	Honeywell HG1700 AG11
4	IMU_HG1700_AG17	Honeywell HG1700 AG17
5	IMU_HG1900_CA29	Honeywell HG1900 CA29
8	IMU_LN200	Litton LN-200 (200 Hz model)
11	IMU_HG1700_AG58	Honeywell HG1700 AG58
12	IMU_HG1700_AG62	Honeywell HG1700 AG62
13	IMU_IMAR_FSAS	iMAR iIMU-FSAS
16	IMU_KVH_COTS	KVH IMU-CPT
19	IMU_LITEF_LCI1	Northrop Grumman Litef LCI-1
20	IMU_HG1930_AA99	Honeywell HG1930 AA99
26	IMU_ISA100C	Northrop Grumman Litef ISA-100C
27	IMU_HG1900_CA50	Honeywell HG1900 CA50
28	IMU_HG1930_CA50	Honeywell HG1930 CA50
31	IMU_ADIS16488	Analog Devices ADIS16488
32	IMU_STIM300	Sensonor STIM300
33	IMU_KVH_1750	KVH1750 IMU
34	IMU_ISA100	Northrop Grumman Litef ISA-100
38	IMU_ISA100_400HZ	Northrop Grumman Litef ISA-100 400 Hz
39	IMU_ISA100C_400HZ	Northrop Grumman Litef ISA-100C 400 Hz
45	IMU_KVH_1725	KVH 1725 IMU
52	IMU_LITEF_MICROIMU	Litef μIMU

Values not shown on this table are reserved.

4.2.5 DUALANTENNAPORTCONFIG Select Dual Antenna Source Port

When the SPAN system is configured for dual antenna, it automatically attempts to connect to an ALIGN capable rover to establish dual antenna corrections. It also attempts to re-establish these corrections should they stop.

For ProPak6 receivers with the Dual Antenna feature, the secondary internal receiver card is used as the ALIGN rover. For other receivers, the default port for connecting to the ALIGN rover is COM2. If an IMU is connected to COM2, COM1 is used instead.

This command is used to designate a different serial port to be used for dual antenna positioning, or to disable this automatic configuration altogether. If automatic configuration is disabled, dual antenna corrections can still be used, but ALIGN corrections must be manually configured.

> Before using this command on a ProPak6 with the Dual Antenna feature, disable the internal dual antenna corrections by sending the command `DUALANTENNACONFIG DISABLE`.

Message ID: 1356

Abbreviated ASCII Syntax:

```
DUALANTENNAPORTCONFIG Port_Selection
```

Abbreviated ASCII Example:

```
DUALANTENNAPORTCONFIG COM3
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	Port_Selection	0	NOPORT	Specify which serial port should be used to communicate with an external ALIGN capable receiver. Selecting NOPORT disables automatic dual antenna configuration.	ENUM	4	H
		1	COM1				
		2	COM2				
		3	COM3				
		19	COM4				
		31	COM5				
		32	COM6				
		34	COM7				
		35	COM8				
		36	COM9				
		37	COM10				

4.2.6 ENCLOSURECOMSELECT
Sets the External COM Port to COM3 or USB

This command enables either COM3 or USB communications on a SPAN-IGM enclosure.

> The `ENCLOSURECOMSELECT` command is for use with only the SPAN-IGM enclosure.
> Do not use this command on other SPAN receivers.

Message ID: **1536**

Abbreviated ASCII Syntax:

```
ENCLOSURECOMSELECT ComSelect
```

Abbreviated ASCII Example:

```
ENCLOSURECOMSELECT USB
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	ComSelect	COM3	0	The COM3 port is enabled on the AUX connector on the SPAN-IGM.	Enum	4	H
		USB	1	The USB port is enabled on the AUX connector on the SPAN-IGM.			

> Issuing the command `ENCLOSURECOMSELECT COM3` disables the EVENT1 input.
> See the OEM6 Family Installation and Operation User Manual (OM-20000128) for more information about user selectable port configurations.

4.2.7 ENCLOSUREWHEELSENSOR Control Wheel Sensor Data

Use this command to enable or disable wheel sensor data from a SPAN-IGM-S1, a SPAN system with an IMU that uses a UIC card, or a SPAN system with an IMU in an IMU Enclosure (IMU-ISA-100C, IMU-ISA100, IMU-HG1900, IMU-LN200 or IMU-µIMU).

> This command must be disabled if the wheel sensor information is received from a source other than the MIC or UIC. For example, the wheel sensor is connected directly to the IMU.

Message ID: 1535

Abbreviated ASCII Syntax:

```
ENCLOSUREWHEELSENSOR switch [Reserved]
```

Abbreviated ASCII Example:

```
ENCLOSUREWHEELSENSOR ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	switch	Disable	0	Disable wheel sensor data. (default)	Enum	4	H
		Enable	1	Enable wheel sensor data.			
3	Reserved	Reserved			Enum	4	H+4

4.2.8 EVENTINCONTROL
Control Event-In Properties

This command controls up to four Event-In input triggers. Each input can be used as either an event strobe or a pulse counter.

When used as an event strobe, an accurate GPS time or position is applied to the rising or falling edge of the input event pulse (See the MARKxTIME or MARKxPOS logs in the OEM6 Family Firmware Reference Manual (OM-20000129) and the MARKxPVA logs in *INS Logs* chapter of this manual). Each input strobe is usually associated with a separate device, therefore different solution output lever arm offsets can be applied to each strobe.

When used as a pulse counter, an internal accumulator is incremented with each input pulse. The accumulated count is output each second using the MARKxCOUNT logs and coordinated with 1PPS. The accumulator begins counting from zero with each new second.

> In some cases a pin on the connector may have more than one function. The function is chosen using the `IOCONFIG` command (see the OEM6 Family Firmware Reference Manual (OM-200000129)).

Message ID: 1637

Abbreviated ASCII Syntax:

```
EVENTINCONTROL mark switch [polarity] [t_bias] [t_guard]
```

ASCII Example:

```
EVENTINCONTROL MARK1 ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	EVENTIN CONTROL header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	mark	MARK1	0	Choose which Event-In Mark to change. This value must be specified. Note: MARK3 and MARK4 are only available on an OEM638 or ProPak6 receiver.	Enum	4	H
		MARK2	1				
		MARK3	2				
		MARK4	3				

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
3	switch	DISABLE	0	Disables the Event Input	Enum	4	H+4
		EVENT	1	Enables the Event Input as an event strobe (default)			
		COUNT	2	Enables the Event Input as a counter. The counter increments with each pulse detected The period of the count is from one PPS to the next PPS. This option is commonly used for a wheel sensor. **Note**: This option functions only on SPAN systems with an OEM638 or ProPak6 receiver.			
		ENABLE	3	A synonym for the EVENT option (for compatibility with previous releases)			
4	polarity	NEGATIVE	0	Negative polarity (default)	Enum	4	H+8
		POSITIVE	1	Positive polarity			
5	t_bias	default: 0 minimum: -999,999,999 maximum: 999,999,999		A constant time bias in nanoseconds can be applied to each event pulse. Typically this is used to account for a transmission delay. This field is not used if the switch field is set to COUNT.	Long	4	H+12
6	t_guard	default: 4 minimum: 2 maximum: 3,599,999		The time guard specifies the minimum number of milliseconds between pulses. This is used to coarsely filter the input pulses. This field is not used if the switch field is set to COUNT.	Ulong	4	H+16

4.2.9 EVENTOUTCONTROL
Control Event-Out Properties

This command is for use with only OEM638 and ProPak6 receivers.

This command configures up to seven Event-Out output strobes. The event strobes toggle between 3.3 V and 0 V. The pulse consists of two periods: one active period followed by a not active period. The start of the active period is synchronized with the top of the GNSS time second and the signal polarity determines whether the active level is 3.3 V or 0 V. The not active period immediately follows the active period and has the alternate voltage.

The outputs that are available vary according to the platform. In some cases, a pin on the connector may have more than one function. The function is chosen using the IOCONFIG command (see the OEM6 Family Firmware Reference Manual (OM-20000129)).

On the OEM638, MARK1 through MARK7 are available. On the ProPak6 only MARK1 through MARK3 are available on the Personality Port.

Message ID: 1636

Abbreviated ASCII Syntax:

```
EVENTOUTCONTROL mark switch [polarity] [active_period]
[non_active_period]
```

ASCII Example:

```
eventoutcontrol mark3 enable
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	EVENTOUT CONTROL header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	mark	MARK1	0	Choose which Event-Out Mark to change. This value must be specified.	Enum	4	H
		MARK2	1				
		MARK3	2				
		MARK4	3				
		MARK5	4				
		MARK6	5				
		MARK7	6				
3	switch	DISABLE	0	Disables the Event Output	Enum	4	H+4
		ENABLE	1	Enables the Event Output (default)			
4	polarity	NEGATIVE	0	Negative polarity (active = 0V) (default)	Enum	4	H+8
		POSITIVE	1	Positive polarity (active = 3.3V)			

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
5	active_period	default: 500,000,000 minimum: 10 maximum: 999,999,990		Active period of the Event Out signal in nanoseconds	Ulong	4	H+12
6	non_active_period	default: 500,000,000 minimum: 10 maximum: 999,999,990		Non-active period of the Event Out signal in nanoseconds	Ulong	4	H+16

4.2.10 EXTERNALPVAS
Enter PVA Update

This command should only be used by advanced users of GNSS+INS.

This command allows a user to provide full position, velocity and attitude updates, in any combination, to the INS. The user can also provide height or attitude only updates, along with Zero Velocity Updates (ZUPTs). These position and velocity updates are entered in local level frame or ECEF.

Message ID: 1463

Abbreviated ASCII Syntax:

```
EXTERNALPVAS Position1 Position2 Position3 Velocity1 Velocity2 Velocity3
Attitude1 Attitude2 Attitude3 PosStdDev1 PosStdDev2 PosStdDev3
VelStdDev1 VelStdDev2 VelStdDev3 AttStdDev1 AttStdDev2 AttStdDev3
UpdateMask OptionsMask
```

Abbreviated ASCII Example:

```
EXTERNALPVAS 51.13495816 -114.03232307 1064.5895 -10.4502 0.2485
-0.09598 1.3152366 -3.6474718 179.5885212 0.01 0.01 0.01 0.01 0.01 0.01
0.1 0.1 0.1 C020 1
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Position1			Latitude in degrees or ECEF X-coordinate in metres	Double	8	H
3	Position2			Longitude in degrees or ECEF Y-coordinate in metres	Double	8	H+8
4	Position3			Height or ECEF Z-coordinate in metres	Double	8	H+16
5	Velocity1			North velocity or velocity along the X-axis in metres/second	Float	4	H+24
6	Velocity2			East velocity or velocity along the Y-axis in metres/second	Float	4	H+28
7	Velocity3			Up velocity or velocity along the Z-axis in metres/second	Float	4	H+32
8	Attitude1			Roll in local level in degrees	Float	4	H+36
9	Attitude2			Pitch in local level in degrees	Float	4	H+40
10	Attitude3			Azimuth in local level in degrees	Float	4	H+44
11	PosStdDev1			Position1 standard deviation in metres	Float	4	H+48
12	PosStdDev2			Position2 standard deviation in metres	Float	4	H+52
13	PosStdDev3			Position3 standard deviation in metres	Float	4	H+56
14	VelStdDev1			Velocity1 standard deviation in metres/second	Float	4	H+60

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
15	VelStdDev2			Velocity2 standard deviation in metres/second	Float	4	H+64
16	VelStdDev3			Velocity3 standard deviation in metres/second	Float	4	H+68
17	AttStdDev1			Attitude1 standard deviation in degrees	Float	4	H+72
18	AttStdDev2			Attitude2 standard deviation in degrees	Float	4	H+76
19	AttStdDev3			Attitude3 standard deviation in degrees	Float	4	H+80
20	UpdateMask			This mask selects which updates are applied. Setting a bit applies the update and more than one update can be applied at one time. See *Table 15, EXTERNALPVAS Updates Mask* on page 48.	HEX Ulong	4	H+84
21	OptionsMask			This mask selects the update options. See *Table 16, EXTERNALPVAS Options Mask* on page 49.	HEX Ulong	4	H+88

Table 15: EXTERNALPVAS Updates Mask

Bit	Mask	Description
0	0x00001	Reserved
1	0x00002	Reserved
2	0x00004	ZUPT Update. No fields required in the `EXTERNALPVAS` command for this update.
3	0x00008	Reserved
4	0x00010	Reserved
5	0x00020	External Position Update. This update is entered using Position1 to Position3 in the `EXTERNALPVAS` command.
6	0x00040	Reserved
7	0x00080	Reserved
8	0x00100	Reserved
9	0x00200	Reserved
10	0x00400	Reserved
11	0x00800	Reserved
12	0x01000	Reserved
13	0x02000	Reserved
14	0x04000	External Velocity Update. This update is entered using Velocity1 to Velocity3 in the `EXTERNALPVAS` command.
15	0x08000	External Attitude Update. This update is entered using Attitude1 to Attitude3 in the `EXTERNALPVAS` command.

Bit	Mask	Description
16	0x10000	External Heading Update. This update is entered using Attitude3 in the `EXTERNALPVAS` command.
17	0x20000	External Height Update. This update is entered using Position3 in the `EXTERNALPVAS` command.

If both the External Position Update and External Height Update bits are set, only the External Position Update will be applied.

If both the External Attitude Update and External Heading Update bits are set, only the External Attitude Update will be applied.

Table 16: EXTERNALPVAS Options Mask

Bit	Mask	Description
0	0x1	If this bit is set, the position and velocity input frame is set to local level. If cleared, the input frame is ECEF.
1	0x2	If this bit is set, the heading update is set relative. If cleared, the heading update is absolute.

4.2.11 EXTHDGOFFSET
Set the Angular Offset

The EXTHDGOFFSET command is used to specify the angular offset from the dual antenna baseline to the SPAN computation frame. It is highly recommended that these offsets be entered by entering a lever arm to both antennas as the measurement errors will be lower. However this command can be used to enter the offsets directly if necessary.

> Typically, `EXTHDGOFFSET` is used as a log. The angular offsets between the dual antenna baseline and the SPAN computation frame are computed internally and can be viewed by logging the EXTHDGOFFSET command (e.g. `log exthdgoffset once`). The message structure of the log is the same as the structure of the command.

Message ID: 1204

Abbreviated ASCII Syntax:

```
EXTHDGOFFSET Heading HeadingSTD [Pitch] [PitchSTD]
```

Abbreviated ASCII Examples:

```
EXTHDGOFFSET 0.5 1.0 -0.23 1.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-		Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary	-	H	0
2	Heading	±180		Angular offset for the azimuth, or heading between the external aiding source and the IMU forward axis (in degrees).	Double	8	H
3	HeadingSTD	0 - 10		Input heading offset standard deviation (in degrees).	Double	8	H+8
4	Pitch	±90		Angular offset for the pitch between the external aiding source and the IMU forward axis (in degrees). Default = 0.0	Double	8	H+16
5	PitchSTD	0 – 10		Input pitch offset standard deviation (in degrees). Default = 0.0	Double	8	H+24

4.2.12 GIMBALSPANROTATION
Gimbal Frame to SPAN Frame Rotation

Use this command to specify a rotational offset between a gimbal mount and the SPAN computation frame. This command must be used if the mount frame and SPAN computation frame do not match after the mapping from SETGIMBALORIENTATION is applied to the mount. See *Chapter 6, Variable Lever Arm* on page 200 for details on frame definitions.

The message format and definitions are identical to those in the VEHICLEBODYROTATION command. The angles must be entered in the SPAN computation frame and the direction of the angles is from the mount to the SPAN computation frame.

Message ID: 1319

Abbreviated ASCII Syntax:

```
GIMBALSPANROTATION XAngle YAngle ZAngle [XUncert] [YUncert] [ZUncert]
```

Abbreviated ASCII Examples:

```
GIMBALSPANROTATION 0 0 90

GIMBALSPANROTATION 0 0 90 0.1 0.1 1.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-		Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary	-	H	0
2	X Angle	±180		Right hand rotation around the SPAN frame X axis in degrees	Double	8	H
3	Y Angle	±180		Right hand rotation around the SPAN frame Y axis in degrees	Double	8	H+8
4	Z Angle	±180		Right hand rotation around the SPAN frame Z axis in degrees	Double	8	H+16
5	X Uncertainty	0 – 45		Uncertainty of X rotation in degrees. Default is 0	Double	8	H+24
6	Y Uncertainty	0 – 45		Uncertainty of Y rotation in degrees. Default is 0	Double	8	H+32
7	Z Uncertainty	0 – 45		Uncertainty of Z rotation in degrees. Default is 0	Double	8	H+40

4.2.13 HEAVEFILTER Enables or Disables Heave Filtering

Use this command to enable or disable the filter used for heave processing.

> To configure the length of the heave filter, use the SETHEAVEWINDOW command (see *SETHEAVEWINDOW* on page 73).

Message ID: 1427

Abbreviated ASCII Syntax:

```
HEAVEFILTER switch
```

Abbreviated ASCII Example:

```
HEAVEFILTER ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	switch	DISABLE	0	Disables the Heave filter.	Enum	4	H
		ENABLE	1	Enables the Heave filter.			

4.2.14 IMUCONFIGURATION
Send one-time configuration data to an IMU

Certain IMUs have a factory default configuration that is incompatible with SPAN. Use this command to have the receiver configure the IMU settings to the values required for SPAN.

> **Use this command only with the IMU-KVH1750 and IMU-KVH1725.**
>
> Most SPAN-supported IMUs are factory configured to work correctly with SPAN, including KVH 17xx series IMUs ordered from NovAtel. If you ordered a KVH 17xx series IMU directly from KVH, or if you have manually reset the configuration on your KVH 17xx series IMU, you must use this command before the IMU will work with SPAN.
>
> If ordering an IMU-KVH-1750 directly from KVH, you must specify commercial part number 01-0349-02.

> The IMU type and IMU port must already be configured before sending this command. See *CONNECTIMU* on page 38.

IMUs configured in this way only require configuration once; this command does not need to be sent each time the receiver is reset.

Message ID: 1745

Abbreviated ASCII Syntax:

```
IMUCONFIGURATION IMUType [Config Options] [Reserved]
```

Abbreviated ASCII Example:

```
IMUCONFIGURATION IMU_KVH_1750 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	IMUType	*Table 14, IMU Type* on page 39		IMU type	Enum	4	H
3	Config Options	0		Configure the KVH 17xx series IMU to the NovAtel required MSYNC,EXT output (default)	Ushort	2	H+4
		1		Configure the KVH 17xx series IMU to 921600 baud rate and the NovAtel required MSYNC,EXT output			
		2		Configure the KVH 17xx series IMU to 460800 baud rate and the NovAtel required MSYNC,EXT output			
4	Reserved	Reserved			Ushort	2	H+6

4.2.15 INPUTGIMBALANGLE Input Gimbal Angles into the Receiver

Use this command to input information about the current mount gimbal angles. Gimbal angles are the angle from the locked mount frame to the current gimbal location. They are input in the mount body frame. See *Chapter 6, Variable Lever Arm* on page 200 for details on frame definitions.

Message ID: 1317

Abbreviated ASCII Syntax:

```
INPUTGIMBALANGLE XAngle YAngle ZAngle [XUncert] [YUncert] [ZUncert]
```

Abbreviated ASCII Examples:

```
INPUTGIMBALANGLE 0.003 -0.1234 12.837

INPUTGIMBALANGLE 0.003 -0.1234 12.837 0.001 0.001 0.005
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	X Angle	±180		Right hand rotation from the locked mount frame X axis to the current gimbal location in degrees.	Double	8	H
3	Y Angle	±180		Right hand rotation from the locked mount frame Y axis to the current gimbal location in degrees.	Double	8	H+8
4	Z Angle	±180		Right hand rotation from the locked mount frame Z axis to the current gimbal location in degrees.	Double	8	H+16
4	X Uncertainty	0 – 180		Uncertainty of X rotation in degrees. Default is 0	Double	8	H+24
5	Y Uncertainty	0 – 180		Uncertainty of Y rotation in degrees. Default is 0	Double	8	H+32
6	Z Uncertainty	0 – 180		Uncertainty of Z rotation in degrees. Default is 0	Double	8	H+40

4.2.16 INSCOMMAND
INS Control Command

Use this command to enable or disable INS. When INS is disabled, no INS position, velocity or attitude is output (however IMU data is still available). Also, INS aiding of tracking reacquisition is disabled. If the command is used to disable INS and then re-enable it, the INS system has to go through its alignment procedure (equivalent to issuing a `RESET` command). See the relevant SPAN User Manual for information about the SPAN alignment procedures.

Message ID: 379

Abbreviated ASCII Syntax:

```
INSCOMMAND action
```

Abbreviated ASCII Example:

```
INSCOMMAND ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Action	RESET	0	Resets the GNSS/INS alignment and restarts the alignment initialization.	Enum	4	H
		DISABLE	1	Disables INS positioning.			
		ENABLE	2	Enables INS positioning where alignment initialization starts again.			
		START_NO_TIME	3	Raw IMU data will begin to flow upon system startup. IMU data collection can begin before the receiver has a GNSS solution. (default)			
		START_FINE_TIME	4	RAWIMU data will only be output after the system reaches FINESTEERING.[a]			

a. For firmware versions prior to 6.600, this is the default behavior.

4.2.17 INSTHRESHOLDS
Change the INS_HIGH_VARIANCE Threshold

The INSTHRESHOLDS command allows you to customize the criteria by which the system reports the inertial solution status. This criteria is used to determine whether the solution status is reported as INS_SOLUTION_GOOD or INS_HIGH_VARIANCE.

This command is especially useful in situations where the system dynamics are known to be challenging or the SPAN system is using a lower grade IMU.

Message ID: 1448

Abbreviated ASCII Syntax:

```
INSTHRESHOLDS ThresholdConfiguration
```

Abbreviated ASCII Example:

```
INSTHRESHOLDS DEFAULT
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Threshold Configuration	DEFAULT	0	Standard INS status threshold settings	Enum	4	H
		LOW	1	Low INS status threshold settings (only checks the Attitude standard deviation)			
		HIGH	2	High INS status threshold settings			
3	Reserved				Double	8	H+4
4	Reserved				Double	8	H+12
5	Reserved				Double	8	H+20

4.2.18 INSWHEELUPDATE Enables or Disables Use of a Wheel Sensor

Use this command to enable or disable wheel sensor functionality. When enabled, measurements from an attached wheel sensor are included in the SPAN solution.

Message ID: 647

Abbreviated ASCII Syntax:

```
INSWHEELUPDATE trigger
```

Abbreviated ASCII Example:

```
INSWHEELUPDATE ENABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	trigger	DISABLE	0	Disables use of wheel sensor functionality.	Enum	4	H
		ENABLE	1	Enables use of wheel sensor functionality. (default)			

4.2.19 INSZUPT
Request Zero Velocity Update

Use this command to manually perform a Zero Velocity Update (ZUPT).

NovAtel's SPAN Technology System does ZUPTs automatically. It is not necessary to use this command under normal circumstances.

> This command should only be used by advanced users of GNSS/INS and only when the system is truly stationary.
>
> Applying a ZUPT while moving will result in severe instability of the solution.

Message ID: 382

Abbreviated ASCII Syntax:

```
INSZUPT
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Reserved This parameter is optional when using abbreviated ASCII syntax.				BOOL	4	H

4.2.20 INSZUPTCONTROL INS Zero Velocity Update Control

Use this command to control whether ZUPTs are performed by the system.

When enabled, ZUPTs allow the INS to reduce its accumulated errors. Typically, the system automatically detects when it is stationary and applies a ZUPT. For certain applications where it is known the system will never be stationary, such as marine or airborne applications, ZUPTs can be disabled altogether.

Message ID: 1293

Abbreviated ASCII Syntax:

```
INSZUPTCONTROL switch
```

Abbreviated ASCII Example:

```
INSZUPTCONTROL DISABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Switch	DISABLE	0	Disable INS zero velocity updates.	Enum	4	H
		ENABLE	1	Enable INS zero velocity updates. (default)			

4.2.21 INTERFACEMODE
Set Interface Type for a Port

Use this command to specify what type of data a particular port on the receiver can transmit and receive. The receive type tells the receiver what type of data to accept on the specified port. The transmit type tells the receiver what kind of data it can generate.

As an example, you could set the receive type on a port to RTCA in order to accept RTCA differential corrections.

It is also possible to disable or enable the generation or transmission of command responses for a particular port. Disabling of responses is important for applications where data is required in a specific form and the introduction of extra bytes may cause problems, for example RTCA, RTCM, RTCMV3 or CMR. Disabling a port prompt is also useful when the port is connected to a modem or other device that will respond with data the receiver does not recognize.

When INTERFACEMODE *port* NONE NONE OFF is set, the specified port is disabled from interpreting any input or output data. Therefore, no commands or differential corrections are decoded by the specified port. Data can be passed through the disabled port and be output from an alternative port using the pass-through logs PASSCOM, PASSXCOM, PASSAUX and PASSUSB. Refer to the OEM6 Family Firmware Reference Manual (OM-20000129) for information on pass-through logging and the COMCONFIG log.

The INTERFACEMODE of the receiver is also configured for the serial port dedicated to the IMU. This mode changes automatically upon sending a CONNECTIMU command and the change is reflected when logging this command. This is normal operation.

> When the CONNECTIMU command (see page 38) is used to configure the IMU connected to the receiver, the correct interface mode for the IMU port is automatically set. The IMU port should not be altered using the INTERFACEMODE command in normal operation. Doing so may result in the loss of IMU communication.

Message ID: 3

Abbreviated ASCII Syntax:

```
INTERFACEMODE [port] rxtype txtype [responses]
```

ASCII Example:

```
INTERFACEMODE COM1 RTCA NOVATEL ON
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Port	See *Table 18, COM Serial Port Identifiers* on page 62		Serial port identifier (default = THISPORT)	Enum	4	H
3	Rxtype	See *Table 17, Serial Port Interface Modes* on page 61		Receive interface mode	Enum	4	H+4
4	Txtype			Transmit interface mode	Enum	4	H+8
5	Responses	OFF	0	Turn response generation off	Enum	4	H+12
		ON	1	Turn response generation on (default)			

Table 17: Serial Port Interface Modes

Binary Value	ASCII Mode Name	Description
0	NONE	The port accepts/generates nothing.
1	NOVATEL	The port accepts/generates NovAtel commands and logs.
2	RTCM	The port accepts/generates RTCM corrections.
3	RTCA	The port accepts/generates RTCA corrections.
4	CMR	The port accepts/generates CMR corrections.
5-6	Reserved	
7	IMU	This port supports communication with a NovAtel supported IMU.
8	RTCMNOCR	This port accepts/generates RTCM with no CR/LF appended. [a]
9-13	Reserved	
14	RTCMV3	The port accepts/generates RTCM Version 3.0 corrections.
15	NOVATELBINARY	The port only accepts/generates binary messages. If an ASCII command is entered when the mode is set to binary only, the command is ignored. Only properly formatted binary messages are responded to and the response is a binary message.
19	IMARIMU	This port supports communication with an iMAR IMU.
20-22	Reserved	
23	KVHIMU	This port supports communication with a KVH CG5100 IMU.
24-26	Reserved	
27	AUTO	For auto-detecting different RTK correction formats.
28	LITEFIMU	This port supports communication with a Litef LCI-1 IMU.
29-34	Reserved	
35	NOVATELX	The port accepts/generates NOVATELX corrections.
36-40	Reserved	
41	KVH1750IMU	This port supports communication with a KVH 17xx series IMU.

a. An output interfacemode of RTCMNOCR is identical to RTCM but with the CR/LF appended. An input interfacemode of RTCMNOCR is identical to RTCM and functions with or without the CR/LF.

Table 18: COM Serial Port Identifiers

Binary	ASCII	Description
1	COM1	COM port 1
2	COM2	COM port 2
3	COM3	COM port 3
6	THISPORT	The current COM port
8	ALL	All COM ports
9	XCOM1 [a]	Virtual COM1 port
10	XCOM2 [a]	Virtual COM2 port
13	USB1 [b]	USB port 1
14	USB2 [b]	USB port 2
15	USB3 [b]	USB port 3
16	AUX	AUX port
19	COM4	COM port 4
23	ICOM1	IP virtual COM port 1
24	ICOM2	IP virtual COM port 2
25	ICOM3	IP virtual COM port 3
31	COM5	COM port 5
32	COM6	COM port 6
34	COM7	COM port 7
35	COM8	COM port 8
36	COM9	COM port 9
37	COM10	COM port 10

a. The XCOM1 and XCOM2 identifiers are not available with the COM command but may be used with other commands. For example, INTERFACEMODE and LOG.

b. The only other field that applies when a USB port is selected is the echo field. Place holders must be inserted for all other fields to use the echo field in this case.

4.2.22 LEVERARMCALIBRATE INS Calibration Command

Use the LEVERARMCALIBRATE command to control the IMU to antenna lever arm calibration.

> `LEVERARMCALIBRATE` is not available for the IMU-CPT, HG-1930, ADIS-16488, IMU-IGM or STIM300.

The IMU to antenna lever arm is the distance from the center of navigation of the IMU to the phase center of the antenna. For information about the IMU center of navigation, refer to the labels on the IMU enclosure or the IMU drawings in the relevant SPAN User Manual. See also *SETIMUTOANTOFFSET* on page 79 and the lever arm calibration routine in the relevant SPAN User Manual.

The calibration runs for the time specified or until the specified uncertainty is met. The BESTLEVERARM log outputs the lever arm once the calibration is complete, see also *BESTLEVERARM* on page 121.

> If a SETIMUANTENNAOFFSET command is already entered (or there is a previously saved lever arm in NVM), before the LEVERARMCALIBRATE is sent, the calibration starts using initial values from SETIMUTOANTOFFSET (or NVM). Ensure the initial standard deviations are representative of the initial lever arm values.

Message ID: 675

Abbreviated ASCII Syntax:

```
LEVERARMCALIBRATE [switch] [maxtime] [maxstd]
```

Abbreviated ASCII Example 1:

```
LEVERARMCALIBRATE ON 600
```

Given this command, the lever arm calibration runs for 600 seconds. The final standard deviation of the estimated lever arm is output in the BESTLEVERARM log.

> The calibration starts when the SPAN solution has converged to an acceptable level. This occurs shortly after the SPAN solution displays INS_SOLUTION_GOOD. The example's 600 second duration is from when calibration begins and not from when you issue the command.

Abbreviated ASCII Example 2:

```
LEVERARMCALIBRATE ON 600 0.05
```

Given this command, the lever arm calibration runs for 600 seconds or until the estimated lever arm standard deviation is <= 0.05 m in each direction (x, y, z), whichever happens first.

Abbreviated ASCII Example 3:

```
LEVERARMCALIBRATE OFF
```

This command stops the calibration. The current estimate, when the command was received, is output in the BESTLEVERARM log, and used in the SPAN computations.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Switch	OFF	0	Enable or disable lever arm calibration.	Enum	4	H
		ON (default)	1				
3	Maxtime	300 - 1800 (default = 600)		Maximum calibration time (s)	Double	8	H+4
4	Maxstd	0.02 – 0.5		Maximum offset uncertainty (m)	Double	8	H+12

4.2.23 NMEATALKER Set the NMEA Talker ID

Use this command to alter the behavior of the NMEA talker ID. The talker is the first 2 characters after the $ sign in the log header of the GPGLL, GPGRS, GPGSA, GPGST, GPGSV, GPRMB, GPRMC and GPVTG log outputs. The other NMEA logs are not affected by the NMEATALKER command.

> The GPGGA position is always based on the position solution from the BESTPOS log which incorporate GNSS + INS solutions as well.

The default GPS NMEA message (`NMEATALKER GP`) outputs GP as the talker ID regardless of the position type given in position logs such as BESTPOS. The `NMEATALKER auto` command switches the talker ID between GP, GN and IN according to the position type given in position logs.

If nmeatalker is set to auto and there are both GPS and GLONASS satellites in the solution, two sentences with the GN talker ID are output. The first sentence contains information about the GPS satellites in the solution and the second sentence contains information about the GLONASS satellites in the solution.

If nmeatalker is set to auto and there are only GLONASS satellites in the solution, the talker ID of this message is GL.

If the solution comes from SPAN, the talker ID is IN.

Message ID: 861

Abbreviated ASCII Syntax:

```
NMEATALKER ID
```

Factory Default:

```
NMEATALKER gp
```

ASCII Example:

```
NMEATALKER auto
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	ID	GP	0	GPS (GP) only	Enum	4	H
		AUTO	1	GPS, Inertial (IN) and/or GLONASS			

> This command only affects NMEA logs capable of a GNSS position output. For example, GPGSV is for information on GNSS satellites and its output always uses the GP ID. *Table 19, NMEA Talkers* shows the NMEA logs and whether they use GP/GN or GP/GN/IN IDs with `NMEATALKER auto`.

Table 19: NMEA Talkers

Log	Talker IDs
GLMLA	GL
GPALM	GP
GPGGA	GP
GPGLL	GP/GN/IN
GPGRS	GP/GN
GPGSA	GP/GN
GPGST	GP/GN/IN
GPGSV	GP/GL
GPRMB	GP/GN/IN
GPRMC	GP/GN/IN
GPVTG	GP/GN/IN
GPZDA	GP

4.2.24 RELINSAUTOMATION Enables Relative INS on the Rover

Use this command to configure the Relative INS plug and play feature on the rover receiver. RELINSAUTOMATION enables/disables the plug and play feature, sets the rover COM port the master receiver is connected to, sets the baud rate for communication, sets the correction transfer rate and enables/ disables sending the HEADINGEXTB/HEADINGEXT2B log back to the master receiver.

On issuing this command at the rover receiver, the rover will automatically sync with the master receiver and configure it to send corrections at the specified baud rate and specified data rate.

> The recommended method for configuring Relative INS is to use the RELINSCONFIG command. See *RELINSCONFIG* on page 69.

> This command should only be issued at the rover receiver.

> If the rover receiver is not connected to the master receiver using a serial COM port, use the `RELINCONFIC` command (see *RELINSCONFIG* on page 69).

Message ID: 1763

Abbreviated ASCII Syntax:

```
RELINSAUTOMATION option [comport] [baudrate] [datarate]
[headingextboption]
```

Abbreviated ASCII Example:

```
RELINSAUTOMATION enable com2 230400 10 on
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	option	ENABLE	0	Enables or disables the plug and play feature.	Enum	4	H
		DISABLE	1				
3	comport	COM1	1	The COM port on the rover receiver to which the master receiver is connected. (default = COM2)	Enum	4	H+4
		COM2	2				
		COM3	3				
4	baudrate	9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600		The baud rate used for communication between the master and rover receivers.	Ulong	4	H+8
5	datarate	1, 2, 4, 5, 10 or 20 Hz		The rate at which corrections are transferred between the receivers. (default =10 Hz)	Ulong	4	H+12

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
6	heading extboption	ON OFF		Enables or disables sending the HEADINGEXTB/HEADINGEXT2B log back to the master receiver. (default = ON)	Enum	4	H+16

4.2.25 RELINSCONFIG Configure Relative INS

Use this command to configure Relative INS on this receiver.

Message ID: 1797

Abbreviated ASCII Syntax:

```
RELINSCONFIG enable rxtype [port] [baud] [rateinhz]
```

Abbreviated ASCII Example:

```
RELINSCONFIG ENABLE ROVER COM2 230400 10
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	enable	DISABLE	0	Enables or disables the Relative INS functionality.	Enum	4	H
		ENABLE	1				
3	rxtype	ROVER	1	Defines the receiver as the master or rover in a Relative INS configuration.	Enum	4	H+4
		MASTER	2				
4	port	See *Table 20, COM Ports* on page 69		Communication port used to communicate with the other receiver. (default = COM2)	Enum	4	H+8
5	baud	9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600		The baud rate used for communication between the master and rover receivers. (default = 230400)	Ulong	4	H+12
6	rateinhz	1, 2, 4, 5, 10 or 20 Hz		The rate at which corrections are transferred between the receivers. (default =10 Hz)	Ulong	4	H+16

Table 20: COM Ports

Binary	ASCII	Description
1	COM1	COM port 1
2	COM2	COM port 2
3	COM3	COM port 3
13	USB1	USB port 1
14	USB2	USB port 2
15	USB3	USB port 3
23	ICOM1	IP virtual COM port 1
24	ICOM2	IP virtual COM port 2
25	ICOM3	IP virtual COM port 3

4.2.26 RVBCALIBRATE
Vehicle to Body Rotation Control

Use the `RVBCALIBRATE` command to enable or disable the calculation of the vehicle to body angular offset. This command should be entered when the IMU is re-mounted in the vehicle or if the rotation angles available are known to be incorrect.

> After the `RVBCALIBRATE ENABLE` command is entered, there are no vehicle body rotation parameters present and a kinematic alignment is NOT possible. Therefore this command should only be entered after the system has performed either a static or kinematic alignment and has a valid INS solution.

A good INS solution and vehicle movement are required for the SPAN system to solve the vehicle-body offset. The solved vehicle body rotation parameters are output in the VEHICLEBODYROTATION log when the calibration is complete, see *page 198*. When the calibration is done, the rotation values are fixed until the calibration is re-run by entering the `RVBCALIBRATE` command again.

> The solved rotation values are used only for a rough estimate of the angular offsets between the IMU and vehicle frames. The offsets are used when aligning the system while in motion (see the start up and alignment technique in the relevant SPAN User Manual). The angular offset values are not applied to the attitude output, unless the `APPLYVEHICLEBODYROTATION` command is enabled.

Message ID: 641

Abbreviated ASCII Syntax:

```
RVBCALIBRATE switch
```

Abbreviated ASCII Example:

```
RVBCALIBRATE enable
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Switch	RESET	0	Control the vehicle/body rotation computation	Enum	4	H
		DISABLE	1				
		ENABLE	2				

4.2.27 SETALIGNMENTVEL
Set the Minimum Kinematic Alignment Velocity

Use the `SETALIGNMENTVEL` command to adjust the minimum required velocity for a kinematic alignment.

Useful in such cases as helicopters, where the alignment velocity should be increased to prevent a poor alignment at low speed.

Message ID: 1397

Abbreviated ASCII Syntax:

```
SETALIGNMENTVEL velocity
```

Abbreviated ASCII Example

```
SETALIGNMENTVEL 5.0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Velocity	Minimum: 1.15 m/s (Default is 5 m/s)		The minimum velocity, in m/s, required to kinematically align.	Double	8	H

4.2.28 SETGIMBALORIENTATION Set the Gimbal Orientation

Use this command to convert Mount Body frame to Mount Computation frame for SPAN. This is done in the same manner as for the IMU. The mapping definitions for SETGIMBALORIENTATION are the same as they are for the SETIMUORIENTATION command (see *Chapter 6, Variable Lever Arm* on page 200 for details on frame definitions). However, unlike an IMU, SPAN is not be able to auto-detect the orientation of the mount used, so this command must be sent to SPAN. If the command is not sent, SPAN will assume a default mapping of 5. If 5 is not the correct mapping, the SPAN system produces bad results.

Message ID: 1318

Abbreviated ASCII Syntax:

```
SETGIMBALORIENTATION mapping
```

Abbreviated ASCII Example:

```
SETGIMBALORIENTATION 6
```

Field	Field Type	Value Range	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	–	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary	–	H	0
2	Mapping[a]	1	Mount X axis is pointing UP	Long	4	H
		2	Mount X axis is pointing DOWN			
		3	Mount Y axis is pointing UP			
		4	Mount Y axis is pointing DOWN			
		5	Mount Z axis is pointing UP (default)			
		6	Mount Z axis is pointing DOWN			

a. See *Table 21, Full Mapping Definitions* on page 76 for details

4.2.29 SETHEAVEWINDOW Set Heave Filter Length

Use this command to control the length of the heave filter. This filter determines the heave (vertical displacement) of the IMU, relative to a long term level surface.

Message ID: 1383

Abbreviated ASCII Syntax:

```
SETHEAVEWINDOW filterlength
```

Abbreviated ASCII Example:

```
SETHEAVEWINDOW 35
```

Field	Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Filter Length	Integer (1 – 300 s) (default = 20 s)		This filter length will be used in the heave filter. Typically, set the filter length to 5 x wave period	Long	4	H

4.2.30 SETIMUORIENTATION Set IMU Orientation

Use this command to specify which of the IMU axis is aligned with gravity. The IMU orientation can be saved using the SAVECONFIG command so that on start-up, the SPAN system does not have to detect the orientation of the IMU with respect to gravity. This is particularly useful for situations where the receiver is powered while in motion.

1. The default IMU axis definitions are:
   Y - forward
   Z - up
   X - out the right hand side
   It is strongly recommended to mount the IMU in this way with respect to the vehicle.
2. Use this command if the system is to be aligned while in motion using the kinematic alignment routine, see the relevant SPAN User Manual for information about alignment routines.

Regardless of system configuration and IMU orientation, the SETIMUORIENTATION and VEHICLEBODYROTATION commands must be sent to the receiver before attempting a kinematic alignment.

The INS filter is reset anytime the SETIMUORIENTATION command is sent, regardless of whether the orientation is changed.

Ensure that all windows, other than the Console, are closed in NovAtel Connect and then use the SAVECONFIG command to save settings in NVM. Otherwise, unnecessary data logging occurs and may overload the system.

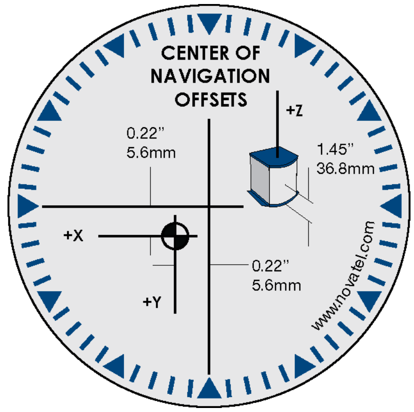
This orientation command serves to transform the incoming IMU signals in such a way that a 5 mapping is achieved (see *Table 21, Full Mapping Definitions* on page 76). For example, if the IMU is mounted with the X axis pointing up and a mapping of 1 is specified then this transformation of the raw IMU data is done:

X⇒Z, Y⇒X, Z⇒Y (where the default is X⇒X, Y⇒Y, Z⇒Z)

Notice that the X-axis observations are transformed into the Z axis, resulting in Z being aligned with gravity and a 5 mapping. The SPAN frame is defined so that Z is always pointing up along the gravity vector. If the IMU mapping is set to 1, the X axis of the IMU enclosure is mapped to the SPAN frame Z axis (pointing up), its Y axis to SPAN frame X and its Z axis to SPAN frame Y.

The X (pitch), Y (roll) and Z (azimuth) directions of the inertial enclosure frame are clearly marked on the IMU. See the relevant SPAN User Manual for the IMU choices and their technical specifications. The example from the LN-200 is shown in *Figure 2, Frame of Reference*.

Figure 2: Frame of Reference



1. Azimuth is positive in a clockwise direction while yaw is positive in a counter-clockwise direction when looking down the axis center. Yaw follows the right-handed system convention where as azimuth follows the surveying convention.
2. The data in the RAWIMU logs is never mapped. The axes referenced in the RAWIMU log descriptions form the IMU enclosure frame (as marked on the enclosure).

Message ID: 567

Abbreviated ASCII Syntax:

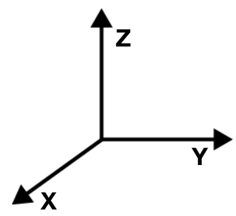
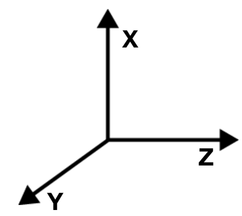
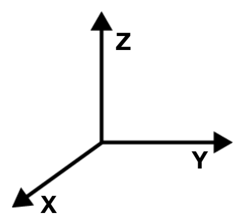
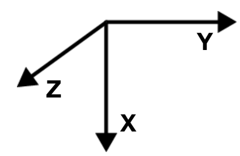
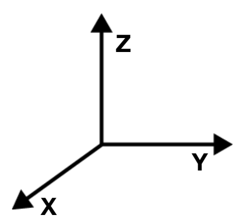
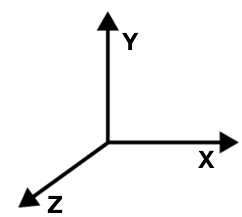
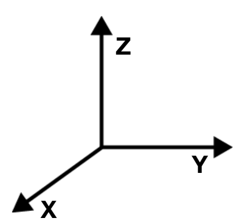
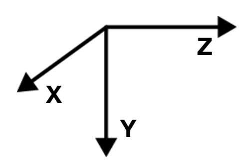
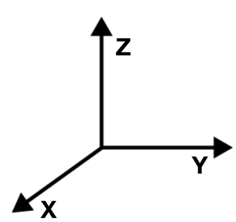
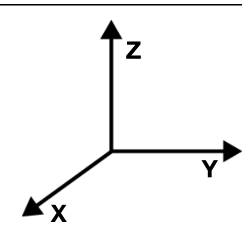
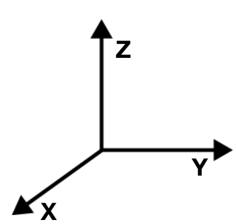
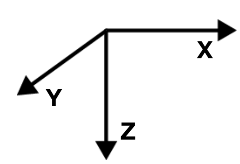
```
SETIMUORIENTATION switch
```

Abbreviated ASCII Example:

```
SETIMUORIENTATION 1
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Switch	0	0	IMU determines axis orientation automatically during coarse alignment. (default)	Long	4	H
		1	1	IMU X axis is pointing UP			
		2	2	IMU X axis is pointing DOWN			
		3	3	IMU Y axis is pointing UP			
		4	4	IMU Y axis is pointing DOWN			
		5	5	IMU Z axis is pointing UP			
		6	6	IMU Z axis is pointing DOWN			

Table 21: Full Mapping Definitions

Mapping	SPAN Frame Axis	SPAN Frame	IMU Enclosure Frame Axis	IMU Enclosure Frame
1	X		Y	
	Y		Z	
	Z		X	
2	X		Z	
	Y		Y	
	Z		-X	
3	X		Z	
	Y		X	
	Z		Y	
4	X		X	
	Y		Z	
	Z		-Y	
5 (default)	X		X	
	Y		Y	
	Z		Z	
6	X		Y	
	Y		X	
	Z		-Z	

4.2.31 SETIMUPORTPROTOCOL
Sets the Protocol Used for the IMU Serial Port

Use the SETIMUPORTPROTOCOL command to force the IMU serial port to use either RS-232 or RS-422 protocol. This overrides the default configured internally when the CONNECTIMU command is sent.

> Before changing the IMU serial port protocol to RS-422, make sure the receiver port connected to the IMU is capable of RS-422 protocol. Refer to the OEM6 Family Installation and Operation User Manual (OM-20000128) or ProPak6 User Manual (OM-20000148) for information about the receiver serial ports.

Message ID: 1767

Abbreviated ASCII Syntax:

```
SETIMUPORTPROTOCOL SerialProtocol
```

Abbreviated ASCII Example:

```
SETIMUPORTPROTOCOL RS422
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Serial Protocol	RS232 RS422		The protocol for the IMU serial port.	Enum	4	H

4.2.32 SETIMUSPECS Specify Error Specifications and Data Rate

Use the `SETIMUSPECS` command to specify the error specifications and data rate for the desired IMU. If the default specs for the supported models are different than the unit used then this command can be used to override the default values.

This command is only available for the following IMUs:

- Honeywell HG1930 (default specifications are for the AA99/CA50 model)
- Honeywell HG1900 (default specifications are for the CA29/CA50 model)

Message ID: 1295

Abbreviated ASCII Syntax:

```
SETIMUSPECS DataRate AccelBias AccelVRW GyroBias GyroARW
AccelSFError GyroSFError [DataLatency]
```

Abbreviated ASCII Example: (iMAR-FSAS Specs)

```
SETIMUSPECS 200 1 .0198 0.75 0.0028 300 300 2.5
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Data Rate	100 Hz to 400 Hz		Data rate of the IMU	Ushort	2	H
3	Accel Bias	-		Total accelerometer bias in milli-g	Double	8	H+2
4	Accel VRW	-		Accelerometer velocity random walk in m/s/rt-hr	Double	8	H+10
5	Gyro Bias	-		Total gyroscope bias in deg/hr	Double	8	H+18
6	Gyro ARW	-		Gyroscope angular random walk in deg/rt-hr	Double	8	H+26
7	Accel Scale Factor Error	> 0		Accelerometer scale factor error in parts per million. Optional. Default = 1000 ppm.	Ulong	4	H+34
8	Gyro Scale Factor Error	> 0		Gyroscopic scale factor error in parts per million. Optional. Default = 1000 ppm.	Ulong	4	H+38
9	Data Latency	> 0		Time delay in milliseconds from the time of validity of the IMU data to the time the input pulse is received by the SPAN enabled receiver. This will include filtering delays, processing delays and transmission times. Optional. Default = 0.0.	Double	8	H+42
10	Reserved	-		Reserved	Ulong	4	H+50
10	CRC	-		32-bit CRC	Hex	4	H+54

4.2.33 SETIMUTOANTOFFSET Set IMU to Antenna Offset

Use this command to enter the offset between the IMU and the GNSS antenna. The measurement should be done as accurately as possible, preferably to within millimeters especially for RTK operation. Any error in the lever arm will translate directly into an error in the INS position.

The x, y and z fields represent the vector from the IMU to the antenna phase center in the IMU enclosure frame. The a, b and c fields are used to enter any possible errors in the measurements. For example, if the 'x' offset measurement accuracy is a centimeter, enter 0.01 in the 'a' field.

Mount the IMU as close as possible to the GNSS antenna, particularly in the horizontal plane.

The X (pitch), Y (roll) and Z (azimuth) directions of the inertial frame are clearly marked on the IMU.

This command should be entered before or during the INS alignment (not after).

After changing the IMU to antenna offset, use the *SAVECONFIG* command to save the changes to non-volatile memory. For information about the *SAVECONFIG* command, see the OEM6 Family Firmware Reference Manual (OM-20000129).

If you are uncertain of the standard deviation values for the antenna offset, err on the side of a larger standard deviation.

Message ID: 383

Abbreviated ASCII Syntax:

```
SETIMUTOANTOFFSET x y z [a] [b] [c]
```

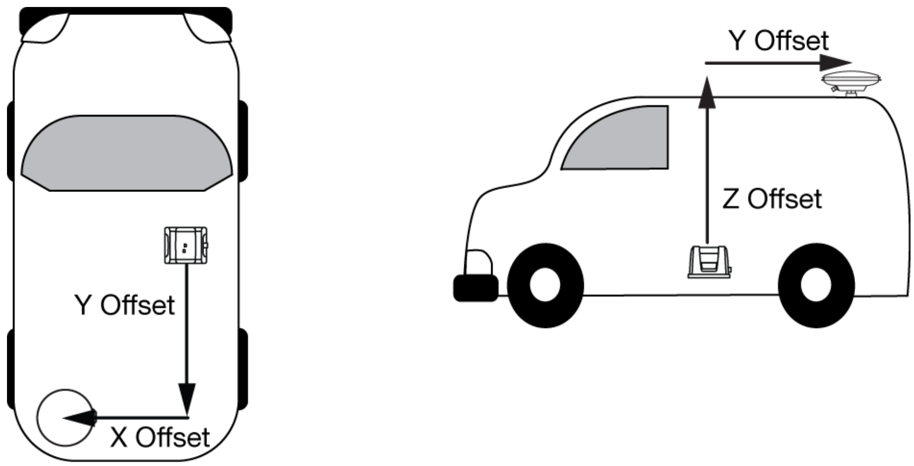
Abbreviated ASCII Example:

```
SETIMUTOANTOFFSET 0.54 0.32 1.20 0.03 0.03 0.05
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	x	±100		x offset (m) See *Figure 3, IMU to Antenna Offset* on page 80	Double	8	H
3	y	±100		y offset (m) See *Figure 3, IMU to Antenna Offset* on page 80	Double	8	H+8
4	z	±100		z offset (m) See *Figure 3, IMU to Antenna Offset* on page 80	Double	8	H+16
5	a	0 to 10		Uncertainty in x (m) (Defaults to 10% of the x offset to a minimum of 0.01 m)	Double	8	H+24
6	b	0 to 10		Uncertainty in y (m) (Defaults to 10% of the y offset to a minimum of 0.01 m)	Double	8	H+32

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
7	c	0 to 10		Uncertainty in z (m) (Defaults to 10% of the z offset to a minimum of 0.01 m)	Double	8	H+40

Figure 3: IMU to Antenna Offset



This example assumes a default mounting configuration and shows a -X offset, -Y offset and +Z offset.

4.2.34 SETIMUTOANTOFFSET2 Set IMU to GNSS2 Antenna Offset

Use the `SETIMUTOANTOFFSET2` command to set the lever arm for the secondary GNSS antenna. Preferably, the primary GNSS antenna is set up behind the IMU forward axis and the secondary GNSS antenna is set up ahead of the IMU forward axis. Entering both lever arms will automatically compute the angular offset between the ALIGN antennas and the IMU axes.

The format of this command is identical to the `SETIMUTOANTOFFSET` command.

Message ID: 1205

Abbreviated ASCII Syntax:

```
SETIMUTOANTOFFSET2 x y z [a] [b] [c]
```

Abbreviated ASCII Example:

```
SETIMUTOANTOFFSET2 0.24 0.32 1.20 0.03 0.03 0.05
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	x	±100		x offset (m) See *Figure 3, IMU to Antenna Offset* on page 80	Double	8	H
3	y	±100		y offset (m) See *Figure 3, IMU to Antenna Offset* on page 80	Double	8	H+8
4	z	±100		z offset (m) See *Figure 3, IMU to Antenna Offset* on page 80	Double	8	H+16
5	a	0 to 10		Uncertainty in x (m) (Defaults to 10% of the x offset to a minimum of 0.01 m)	Double	8	H+24
6	b	0 to 10		Uncertainty in y (m) (Defaults to 10% of the y offset to a minimum of 0.01 m)	Double	8	H+32
7	c	0 to 10		Uncertainty in z (m) (Defaults to 10% of the z offset to a minimum of 0.01 m)	Double	8	H+40

4.2.35 SETIMUTOGIMBALOFFSET Set IMU to Gimbal Mount Offset

Use this command to enter the offset between the IMU and the gimbal mount. The measurement should be done as accurately as possible, preferably to within millimeters, especially for RTK operation. The x, y and z fields represent the vector from the IMU to the gimbal center of navigation. The X Std Dev, Y Std Dev and Z Std Dev fields allow you to enter any possible errors in your measurements. This command is entered in the IMU Body Frame (IMU Enclosure axis frame).

Message ID: Message ID: 1352

Abbreviated ASCII Syntax:

```
SETIMUTOGIMBALOFFSET X Y Z <X Std Dev> <Y Std Dev> <Z Std Dev>
```

Abbreviated ASCII Example:

```
SETIMUTOGIMBALOFFSET 0.5 -0.3 -0.7 0.01 0.01 0.01
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	x	±100		X offset (m)	Double	8	H
3	Y	±100		Y offset (m)	Double	8	H+8
4	Z	±100		Z offset (m)	Double	8	H+16
5	X Std Dev	0 to +10		Uncertainty in x (m) (Defaults to 10% of the x offset to a minimum of 0.01m)	Double	8	H+24
6	Y Std Dev	0 to +10		Uncertainty in y (m) (Defaults to 10% of the y offset to a minimum of 0.01m)	Double	8	H+32
7	Z Std Dev	0 to +10		Uncertainty in z (m) (Defaults to 10% of the z offset to a minimum of 0.01m)	Double	8	H+40

4.2.36 SETINITATTITUDE Set Initial Attitude of SPAN in Degrees

Use this command to input a known attitude to start SPAN operation, rather than the usual coarse alignment process. The caveats and special conditions of this command are listed below:

- This alignment is instantaneous based on the user input. This allows for faster system startup however, the input values must be accurate or SPAN does not perform well.
- If uncertain about the standard deviation of the angles to enter, err on the side of a larger standard deviation.
- Sending `SETINITATTITUDE` resets the SPAN filter. The alignment is instantaneous, but some time and vehicle dynamics are required for the SPAN filter to converge. Bridging performance is poor before filter convergence.
- The roll (about the Y-axis), pitch (about the X-axis) and azimuth (about the Z-axis) are with respect to the SPAN frame. If the IMU enclosure is mounted with the z-axis pointing upward, the SPAN frame is the same as the markings on the enclosure. If the IMU is mounted in another way, SPAN transforms the SPAN frame axes such that Z points up for SPAN computations. You must enter the angles in SETINITATTITUDE with respect to the transformed axis. See *SETIMUORIENTATION* on page 74 for a description of the axes mapping that occurs when the IMU is mounted differently from Z up.
- This command is not save configurable and, if needed, must be entered after the system reports INS_ALIGNING.

> 1. Azimuth is positive in a clockwise direction when looking towards the Z-axis origin.
> 2. Use the `SETIMUORIENTATION` command if the IMU is mounted with the Z-axis not pointing up. Then use the tables in *SETIMUORIENTATION* on page 74 to determine the azimuth axis that SPAN is using.

Message ID: 862

Abbreviated ASCII Syntax:

```
SETINITATTITUDE pitch roll azimuth pitchSTD rollSTD azSTD
```

Abbreviated ASCII Example:

```
SETINITATTITUDE 0 0 90 5 5 5
```

In this example, the initial roll and pitch is set to zero degrees, with a standard deviation of 5 degrees for both. This means that the SPAN system is very close to level with respect to the local gravity field. The azimuth is 90 degrees, also with a 5 degree standard deviation.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Pitch	±360		Input pitch angle, about the X-axis (degrees)	Double	8	H
3	Roll	±360		Input roll angle, about the Y-axis (degrees)	Double	8	H+8
4	Azimuth	±360		Input azimuth angle, about the Z-axis (degrees)	Double	8	H+16

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
5	PitchSTD	0.0002778[a] to 45		Input pitch standard deviation (STD) angle (degrees)	Float	4	H+24
6	RollSTD	0.0002778[a] to 45		Input roll STD angle (degrees)	Float	4	H+28
7	AzSTD	0.0002778[a] to 45		Input azimuth STD angle (degrees)	Float	4	H+32

a. 0.0002778° is equal to 1 arc second.

4.2.37 SETINITAZIMUTH Set Initial Azimuth and Standard Deviation

Use this command to start SPAN operation with a previously known azimuth. Azimuth is the weakest component of a coarse alignment and is also the easiest to know from an external source (i.e., like the azimuth of roadway). When using this command, SPAN operation through alignment will appear the same as with a usual coarse alignment. Roll and pitch is determined using averaged gyro and accelerometer measurements. The input azimuth is used rather than what is computed by the normal coarse alignment routine.

- This alignment takes the same amount of time as the usual coarse alignment (30 s nominally).
- Input azimuth values must be accurate for good system performance.
- Sending `SETINITAZIMUTH` resets the SPAN filter. The re-alignment may take up to 30 seconds, after which some time and vehicle dynamics are still required for the SPAN filter to converge. Bridging performance is poor before filter convergence.
- The azimuth angle is with respect to the SPAN frame. If the IMU enclosure is mounted with the Z-axis pointing upward, the SPAN frame is the same as what is marked on the enclosure. If the IMU is mounted in another way, SPAN transforms the SPAN frame axes such that Z points up for SPAN computations. Enter the azimuth with respect to the transformed axis. See *SETIMUORIENTATION* on page 74 for a description of the axes mapping that occurs when the IMU is mounted differently from Z pointing up.
- This command is not save configurable and, if needed, must be entered after the system reports INS_ALIGNING.

1. Azimuth is positive in a clockwise direction when looking towards the z-axis origin.
2. You do not have to use the SETIMUORIENTATION command, see page 74, unless you have your IMU mounted with the Z-axis not pointing up. Then, use the tables in the SETIMUORIENTATION command to determine the azimuth axis that SPAN is using.

Message ID: 863

Abbreviated ASCII Syntax:

```
SETINITAZIMUTH azimuth azSTD
```

Abbreviated ASCII Example:

```
SETINITAZIMUTH 90 5
```

In this example, the initial azimuth has been set to 90 degrees. This means that the SPAN system Y-axis is pointing due East, within a standard deviation of 5 degrees. Note, if the SPAN system is mounted with the positive Z-axis (as marked on the enclosure) in a direction that is not up, refer to the `SETIMUORIENTATION` command to determine the SPAN computation frame axes mapping that SPAN automatically applies.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary	-	H	0
2	azimuth	±360		Input azimuth angle (degrees)	Double	8	H
3	azSTD	0.0002778[a] to 45		Input azimuth standard deviation angle (degrees)	Float	4	H+8

a. 0.0002778 degrees is equal to 1 arc second.

4.2.38 SETINSOFFSET Set INS Offset

Use the `SETINSOFFSET` command to specify an offset from the IMU for the output position and velocity of the INS solution. This command shifts the position and velocity in the INSPOS, INSPOSS, INSPOSX, INSVEL, INSVELS, INSVELX, INSSPD, INSSPDS, INSPVA, INSPVAS and INSPVAX logs by the amount specified in metres with respect to the IMU enclosure frame axis.

Message ID: 676

Abbreviated ASCII Syntax:

```
SETINSOFFSET xoffset yoffset zoffset
```

Abbreviated ASCII Example:

```
SETINSOFFSET 0.15 0.15 0.25
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	X offset	±100		Offset along the IMU enclosure frame X-axis (m)	Double	8	H
3	Y offset	±100		Offset along the IMU enclosure frame Y-axis (m)	Double	8	H+8
4	Z offset	±100		Offset along the IMU enclosure frame Z-axis (m)	Double	8	H+16

4.2.39 SETINSROTATION Set the Rotation of the Attitude Output

Use this command to rotate the attitude output of the INS solution independently from any kinematic alignment requirements.

The body frame is nominally the frame as marked on the IMU enclosure. If the IMU is not mounted with the Z-axis approximately up, check the new computational axis orientation that SPAN automatically uses, called the SPAN computational frame. SPAN forces the Z-axis to be up in the SPAN computational frame. Output attitude (in logs such as INSPVA and INSATT) is with respect to the SPAN computational frame. Refer to the *SETIMUORIENTATION* on page 74 to see which mapping definition applies, depending on which IMU axis most closely aligns to gravity. Essentially, this means that if the IMU is not mounted with the Z-axis approximately up (as marked on the enclosure), a new IMU frame defines which mapping applies. This new computational frame will not match what is marked on the IMU enclosure and needs to be determined by checking *Table 21, Full Mapping Definitions* on page 76.

With the default mapping and no angular offset between the output frame and SPAN computational frame, the output roll is the angle of rotation about the Y-axis, the output pitch is about the X-axis and the output azimuth is about the Z-axis and is measured to the Y-axis. Note that azimuth is positive in the clockwise direction when looking towards the origin. However, the input vehicle to body rotation about the Z-axis follows the right hand rule convention and a positive rotation is in the counterclockwise direction when looking towards the origin. For further information about extracting the vehicle's attitude with respect to the local level frame, refer to NovAtel application note *APN-037 Application Note on Vehicle Body Rotations*, available from the NovAtel Web site at www.novatel.com/support/search/items/Application%20Note.

> This command overrides the output rotation caused by a `VEHICLEBODYROTATION`/ `APPLYVEHICLEBODYROTATION` command set.
>
> Unlike the `VEHICLEBODYROTATION` command which is specified as the rotation from the vehicle frame to the SPAN frame, this command rotates from the SPAN frame to the desired output frame, thus the sign convention is reversed between the two commands.

Message ID: 1796

Abbreviated ASCII Syntax:

```
SETINSROTATION alpha beta gamma
```

Abbreviated ASCII Example:

```
SETINSROTATION 0 0 90
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	alpha (X-angle)	±180		Right hand rotation about the SPAN computation frame X-axis (degrees)	Double	8	H
3	beta (Y-angle)	±180		Right hand rotation about the SPAN computation frame Y-axis (degrees)	Double	8	H+8
4	gamma (Z-angle)	±180		Right hand rotation about the SPAN computation frame Z-axis (degrees)	Double	8	H+16

4.2.40 SETINSUPDATE
Enable/Disable INS Filter Updates

> This command should only be used by advanced users of GNSS+INS.

Use this command to enable or disable the available INS filter updates.

Message ID: 1821

Abbreviated ASCII Syntax:

```
SETINSUPDATE INSUpdate Trigger
```

Abbreviated ASCII Example:

```
SETINSUPDATE ZUPT DISABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	INSUpdate	POS	0	Position updates	Enum	4	H
		ZUPT	1	Zero velocity updates			
		PSR	2	Pseudorange updates			
		ADR	3	Carrier phase updates			
		DOPPLER	4	Doppler updates			
		ALIGN	5	Heading updates			
		DMI	6	Distance measuring instrument (wheel sensor) updates			
3	Trigger	DISABLE	0	Disable the INS update specified in the INSUpdate field.	Enum	4	H+4
		ENABLE	1	Enable the INS update specified in the INSUpdate field.			

4.2.41 SETMARK1OFFSET Set Mark1 Offset

Use this command to set the offset to the Mark1 trigger event.

Message ID: 1069

Abbreviated ASCII Syntax:

```
SETMARK1OFFSET xoffset yoffset zoffset αoffset βoffset γoffset
```

Abbreviated ASCII Example:

```
SETMARK1OFFSET -0.324 0.106 1.325 0 0 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	x offset	±100		Offset along the IMU enclosure frame X-axis (m) for Mark1	Double	8	H
3	y offset	±100		Offset along the IMU enclosure frame Y-axis (m) for Mark1	Double	8	H+8
4	z offset	±100		Offset along the IMU enclosure frame Z-axis (m) for Mark1	Double	8	H+16
5	αoffset	±180		Roll offset for Mark1 (degrees)[a]	Double	8	H+24
6	βoffset	±180		Pitch offset for Mark1 (degrees)[a]	Double	8	H+32
7	γoffset	±180		Azimuth offset for Mark1 (degrees)[a]	Double	8	H+40

a. Enter the roll, pitch and azimuth offsets in the SPAN computation frame.

4.2.42 SETMARK2OFFSET
Set Mark2 Offset

Use this command to set the offset to the Mark2 trigger event.

> This command is not available for the SPAN-CPT.

Message ID: 1070

Abbreviated ASCII Syntax:

```
SETMARK2OFFSET xoffset yoffset zoffset αoffset βoffset γoffset
```

Abbreviated ASCII Example:

```
SETMARK2OFFSET -0.324 0.106 1.325 0 0 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	x offset	±100		Offset along the IMU enclosure frame X-axis (m) for Mark2	Double	8	H
3	y offset	±100		Offset along the IMU enclosure frame Y-axis (m) for Mark2	Double	8	H+8
4	z offset	±100		Offset along the IMU enclosure frame Z-axis (m) for Mark2	Double	8	H+16
5	αoffset	±180		Roll offset for Mark2 (degrees)[a]	Double	8	H+24
6	βoffset	±180		Pitch offset for Mark2 (degrees)[a]	Double	8	H+32
7	γoffset	±180		Azimuth offset for Mark2 (degrees)[a]	Double	8	H+40

a. Enter the roll, pitch and azimuth offsets in the SPAN computation frame.

4.2.43 SETMARK3OFFSET Set Mark3 Offset

Use this command to set the offset to the Mark3 trigger event.

> This command is only available for SPAN systems with an OEM638 or ProPak6 receiver.

Message ID: **1116**

Abbreviated ASCII Syntax:

```
SETMARK3OFFSET xoffset yoffset zoffset αoffset βoffset γoffset
```

Abbreviated ASCII Example:

```
SETMARK3OFFSET -0.324 0.106 1.325 0 0 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	x offset	±100		Offset along the IMU enclosure frame X-axis (m) for Mark3	Double	8	H
3	y offset	±100		Offset along the IMU enclosure frame Y-axis (m) for Mark3	Double	8	H+8
4	z offset	±100		Offset along the IMU enclosure frame Z-axis (m) for Mark3	Double	8	H+16
5	αoffset	±180		Roll offset for Mark3 (degrees)[a]	Double	8	H+24
6	βoffset	±180		Pitch offset for Mark3 (degrees)[a]	Double	8	H+32
7	γoffset	±180		Azimuth offset for Mark3 (degrees)[a]	Double	8	H+40

a. Enter the roll, pitch and azimuth offsets in the SPAN computation frame.

4.2.44 SETMARK4OFFSET Set Mark4 Offset

Use this command to set the offset to the Mark4 trigger event.

> This command is only available for SPAN systems with an OEM638 or ProPak6 receiver.

Message ID: 1117

Abbreviated ASCII Syntax:

```
SETMARK4OFFSET xoffset yoffset zoffset αoffset βoffset γoffset
```

Abbreviated ASCII Example:

```
SETMARK4OFFSET -0.324 0.106 1.325 0 0 0
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	x offset	±100		Offset along the IMU enclosure frame X-axis (m) for Mark4	Double	8	H
3	y offset	±100		Offset along the IMU enclosure frame Y-axis (m) for Mark4	Double	8	H+8
4	z offset	±100		Offset along the IMU enclosure frame Z-axis (m) for Mark4	Double	8	H+16
5	αoffset	±180		Roll offset for Mark4 (degrees)[a]	Double	8	H+24
6	βoffset	±180		Pitch offset for Mark4 (degrees)[a]	Double	8	H+32
7	γoffset	±180		Azimuth offset for Mark4 (degrees)[a]	Double	8	H+40

a. Enter the roll, pitch and azimuth offsets in the SPAN computation frame.

4.2.45 SETMAXALIGNMENTTIME
Set a Time Limit for Static Course Alignment

Use this command to set a maximum time limit allowed for static coarse alignments. Coarse alignments typically take under 60 seconds, but in heavy vibration conditions they can take much longer trying to compensate for the vibration induced noise. This command is used to cap the time to a specific length.

> This command is for advanced users only.
> Alignment accuracy cannot be guaranteed if the alignment time is capped using this command.

Message ID: 1800

Abbreviated ASCII Syntax:

```
SETMAXALIGNMENTTIME switch [duration]
```

Abbreviated ASCII Example:

```
SETMAXALIGNMENTTIME ENABLE 90
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	switch	DISABLE	0	Disables the static alignment time limit.	Enum	4	H
		ENABLE	1	Enables the static alignment time limit.			
3	duration	30 - 300		Maximum static alignment time in seconds. Default is 180.	Ulong	4	H+4

4.2.46 SETRELINSOUTPUTFRAME Sets the Relative INS Output Frame

Use this command to change the frame of the output solution provided in the RELINSPVA and SYNCRELINSPVA logs. See *Section 5.2.45, RELINSPVA* on page 185 and *Section 5.2.47, SYNCRELINSPVA* on page 188 for information about these logs.

See *Chapter 7, Relative INS* on page 205 for information about the Relative INS functionality.

Message ID: 1775

Abbreviated ASCII Syntax:

```
SETRELINSOUTPUTFRAME OutputFrame [DiffCriteria]
```

Abbreviated ASCII Example:

```
SETRELINSOUTPUTFRAME ECEF TRUE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	OutputFrame	ROVER	1	Frame of the output solution in the RELINSPVA and SYNCRELINSPVA logs. ROVER – the output frame of the Rover INS solution. MASTER – the output frame of the Master INS solution. ECEF – Earth Center Earth Fixed LOCALLEVEL – Local level The default is the ROVER.	Enum	4	H
		MASTER	2				
		ECEF	3				
		LOCALLEVEL	4				
3	DiffCriteria	FALSE	0	The delta solution is computed as Rover minus Master. (default)	Bool	1	H+4
		TRUE	1	The delta solution is computed as Master minus Rover.			

4.2.47 SETUPSENSOR Add a new sensor object

Use this command to add a new sensor object to the system. A sensor object consists of an ID, an Event_Out line and an Event_In line. This is an intended as a simplified way to set up triggering to and from a sensor rather than configuring all connections independently. It also allows for event pulses to be sent to a sensor at specific GPS times (see the TIMEDEVENTPULSE command on page 101).

Message ID: 1333

Abbreviated ASCII Syntax:

```
SETUPSENSOR SensorID EventOut OPP OAP EventIn EIC IPP ITB MITG
```

Abbreviated ASCII Example:

```
SETUPSENSOR SENSOR3 MARK1 POSITIVE 2 MARK4 EVENT POSITIVE 0 2
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Sensor ID	SENSOR1	0	The sensor to configure.	Enum	4	H
		SENSOR2	1				
		SENSOR3	2				
3	EventOut	MARK1	0	Associate a specific MARK Event_Out line to this sensor configuration.	Enum	4	H+4
		MARK2	1				
		MARK3	2				
		MARK4	3				
4	OPP	NEGATIVE	0	Mark output pulse polarity	Enum	4	H+8
		POSITIVE	1				
5	OAP	2 - 500		Mark output active period in milliseconds. Value must be divisible by 2.	Ulong	4	H+12
6	EventIn	MARK1	0	Associate a specific MARK Event_In line to this sensor configuration.	Enum	4	H+16
		MARK2	1				
		MARK3	2				
		MARK4	3				
7	EIC	DISABLE	0	Event in control	Enum	4	H+20
		EVENT	1				
8	IPP	NEGATIVE	0	Mark input pulse polarity	Enum	4	H+24
		POSITIVE	1				
9	ITB	-99999999 to 99999999		Mark input time bias in milliseconds	Long	4	H+28

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
10	ITG	2 to 3599999		Mark input time guard in milliseconds	Ulong	4	H+32

Only MARK1 is available for the SPAN-CPT.

MARK3 and MARK4 are available only on SPAN systems with an OEM638 or ProPak6 receiver.

4.2.48 SETWHEELPARAMETERS
Set Wheel Parameters

The SETWHEELPARAMETERS command can be used when wheel sensor data is available. It gives the filter a good starting point for the wheel size scale factor. It also gives the SPAN filter an indication of the expected accuracy of the wheel data.

Message ID: 847

Abbreviated ASCII Syntax:

```
SETWHEELPARAMETERS ticks circ spacing
```

Abbreviated ASCII Example:

```
SETWHEELPARAMETERS 58 1.96 0.025
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Ticks	1-10000		Number of ticks per revolution	Ushort	4[a]	H
3	Circ	0.1-100		Wheel circumference (m) (default = 1.96 m)	Double	8	H+4
4	Spacing	0.001-1000		Spacing of ticks, or resolution of the wheel sensor (m)	Double	8	H+12

a. In the binary log case, an additional 2 bytes of padding are added to maintain 4 byte alignment.

Fields 2, 3 and 4 do not have to 'add up'. Field 4 is used to weight the wheel sensor measurement. Fields 2 and 3 are used with the estimated scale factor to determine the distance travelled.

4.2.49 SETWHEELSOURCE
Set Wheel Sensor Input Source

This command is only available for SPAN systems with an OEM638 or ProPak6 receiver.

Use the SETWHEELSOURCE command to specify how the wheel sensor is connected to the SPAN system.

If the wheel sensor is connected externally using an EVENT line, the MARK option is used. For example, if the wheel sensor is connected to the EVENT3 line with a negative polarity tick:

```
SETWHEELSOURCE MARK3 NEGATIVE
```

If the wheel sensor is connected directly to the IMU, the IMU option is used.

```
SETWHEELSOURCE IMU
```

The wheel sensor can be connected directly to:

- an IMU that uses the UIC card
- an IMU in an IMU Enclosure (IMU-ISA-100C, IMU-ISA100, IMU-HG1900, IMU-LN200 or IMU-µIMU)
- an iMAR IMU-FSAS
- a SPAN-IGM
- an IMU-IGM

Message ID: 1722

Abbreviated ASCII Syntax:

```
SETWHEELSOURCE input polarity
```

Abbreviated ASCII Example:

SETWHEELSOURCE MARK1 POSITIVE

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	input	IMU	0	Specify to which wheel sensor input the command should be applied. Default is IMU	Enum	4	H
		MARK1	1				
		MARK2	2				
		MARK3	3				
		MARK4	4				
3	polarity	NEGATIVE	0	Specify the polarity of the pulse to be received on the Mark input. (optional) This field is not used if the input is set to IMU. Default is Positive	Enum	8	H+4
		POSITIVE	1				
4	Reserved	-	-	Reserved	Enum	12	H+12

> Only MARK1 is available for the SPAN-CPT.
> MARK3 and MARK4 are available only on SPAN systems with an OEM638 or ProPak6 receiver.

4.2.50 TAGNEXTMARK
Tags the Next Incoming Mark Event

Use this command to tag the next incoming mark event on the selected mark with a 32-bit number. This is available in the TAGGEDMARKxPVA log (see page 190 to page 193) to easily associate the PVA log with a supplied event.

Message ID: 1257

Abbreviated ASCII Syntax:

```
TAGNEXTMARK Mark Tag
```

Abbreviated ASCII Example:

```
TAGNEXTMARK MARK1 1234
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Mark	MARK1	0	Event line	Enum	4	H
		MARK2	1				
		MARK3	2				
		MARK4	3				
3	Tag	-	-	Tag for next mark event	Ulong	4	H+4

> Only MARK1 is available for the SPAN-CPT.
>
> MARK3 and MARK4 are available only on SPAN systems with an OEM638 or ProPak6 receiver.

4.2.51 TIMEDEVENTPULSE
Add a new camera event

Use this command to add a new camera event to the system. TIMEDEVENTPULSE sends a pulse on the sensor MARK output at the selected GPS time and sets the trigger on the sensor MARK input to be tagged with an event ID (see TAGGEDMARK1PVA on page 190 through TAGGEDMARK4PVA on page 193). The lines connected to each sensor are configured using the SETUPSENSOR command (see page 95).

> A maximum of 10 unprocessed events can be buffered into the system. A `TIMEDEVENTPULSE` command must be entered at least 1 second prior to the requested event time.

> The TIMEDEVENTPULSE functionality is available only on OEM638 and ProPak6 receivers.

Message ID: 1337

Abbreviated ASCII Syntax:

```
TIMEDEVENTPULSE SensorID GPSWeek GPSSeconds [Event ID]
```

Abbreviated ASCII Example:

```
TIMEDEVENTPULSE -1 1617 418838 100
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Sensor ID	ALL	-1 (0xFFFFFFFF)	The sensor(s) affected by the trigger command.	Long	4	H
		SENSOR1	0x01	The decimal representation of the combination of bits 0-2 can be used to select a combination of active sensors (e.g. 5 [101] will select sensors 1 and 3).			
		SENSOR2	0x02				
		SENSOR3	0x04				
3	GPS Week	0 - MAX Ulong		The GPS week that triggers the event.	Ulong	4	H+4
4	GPS Seconds	0 - 604800		The GPS week seconds that triggers the event.	Double	8	H+8
5	Event ID	0- MAX Ulong		The event's identifier, used to tag the TAGGEDMARKxPVA logs if a sensor input is enabled. Optional Default = 0	Ulong	4	H+16

4.2.52 VEHICLEBODYROTATION Vehicle to SPAN Frame Rotation

Use the `VEHICLEBODYROTATION` command to set angular offsets between the vehicle frame (direction of travel) and the SPAN body frame (direction that the IMU computational frame is pointing). If the angular offsets are estimated using the `RVBCALIBRATE` command, the `VEHICLEBODYROTATION` command values are used as the initial values. The uncertainty values are optional (defaults = 0.0). See the relevant SPAN User Manual for information about reference frames within SPAN and vehicle to SPAN frame angular offset calibration routines. See *RVBCALIBRATE* on page 70 for more information.

> The body frame is nominally the frame as marked on the IMU enclosure. If the IMU is not mounted with the Z-axis approximately up, check the new computational axis orientation that SPAN automatically uses, called the SPAN computational frame. SPAN forces Z to be up in the SPAN computational frame. Output attitude (in INSPVA, INSATT, etc.) is with respect to the SPAN computational frame. Refer to the `SETIMUORIENTATION` command description to see what mapping definition applies, depending on which IMU axis most closely aligns to gravity. Essentially, this means that if the IMU is not mounted with the Z-axis approximately up (as marked on the enclosure), a new IMU frame defines what mapping applies. This new computational frame will not match what is marked on the IMU enclosure and will need to be determined by checking *Full Mapping Definitions* on page 76.
>
> With the default mapping and with no angular offset between the vehicle frame and SPAN computational frame, the output roll is the angle of rotation about the Y-axis, the output pitch is about the X-axis and the output azimuth is about the Z-axis and is measured to the Y-axis. Note that azimuth is positive in the clockwise direction when looking towards the origin. However, the input vehicle to body rotation about the Z-axis follows the right hand rule convention and a positive rotation is in the counterclockwise direction when looking towards the origin.
>
> For further information about extracting the vehicle's attitude with respect to the local level frame, refer to NovAtel application note APN-037 Application Note on Vehicle Body Rotations, available from the NovAtel Web site at www.novatel.com/support/search/.

> Regardless of system configuration and IMU orientation, the `VEHICLEBODYROTATION` and `SETIMUORIENTATION` commands must be sent to the receiver before attempting a kinematic alignment.

The rotation values are used during kinematic alignment. The rotation is used to transform the vehicle frame attitude estimates from GNSS into the SPAN frame of the IMU during the kinematic alignment. If you use the `APPLYVEHICLEBODYROTATION` command on page 36, the reported attitude is in the vehicle frame; otherwise, the reported attitude is in the SPAN frame. The uncertainty values report the accuracy of the angular offsets.

The VEHICLEBODYROTATION command sets the initial estimates for the angular offset. The uncertainty values are optional.

> Enter rotation angles in degrees. NovAtel recommends entering `SETIMUORIENTATION` first then `VEHICLEBODYROTATION`.

Message ID: 642

Abbreviated ASCII Syntax:

```
VEHICLEBODYROTATION alpha beta gamma [δalpha] [δbeta] [δgamma]
```

Abbreviated ASCII Example:

```
VEHICLEBODYROTATION 0 0 90 0 0 5
```

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	X Angle	±180		Right hand rotation about vehicle frame X-axis (degrees)	Double	8	H
3	Y Angle	±180		Right hand rotation about vehicle frame Y-axis (degrees)	Double	8	H+8
4	Z Angle	±180		Right hand rotation about vehicle frame Z-axis (degrees)	Double	8	H+16
5	X Uncertainty	0 - 45		Uncertainty of X rotation (degrees) (default = 0)	Double	8	H+24
6	Y Uncertainty	0 - 45		Uncertainty of Y rotation (degrees) (default = 0)	Double	8	H+32
7	Z Uncertainty	0 - 45		Uncertainty of Z rotation (degrees) (default = 0)	Double	8	H+40
8	xxxx	-	-	32-bit CRC	Hex	4	H+48
9	[CR][LF]	-	-	Sentence Terminator (ASCII only)	-	-	-

4.2.53 WHEELVELOCITY
Wheel Velocity for INS Augmentation

Use the WHEELVELOCITY command to input wheel sensor data into the OEM6 receiver.

> This command should be used only if the wheel sensor cannot be directly connected to an odometer port in the SPAN system.

> When wheel sensor data is entered using this command, only the *Cumulative Ticks/s* value is used by the system. Values entered for *Wheel Velocity* and *Float Wheel Velocity* are not used at this time.

Message ID: 504

Abbreviated ASCII Example:

```
WHEELVELOCITY 123 8 10 0 0 0 0 40

WHEELVELOCITY 123 8 10 0 0 0 0 80

WHEELVELOCITY 123 8 10 0 0 0 0 120
```

The examples above are for a vehicle traveling at a constant velocity with these wheel sensor characteristics:

- Wheel Circumference = 2 m
- Vehicle Velocity (assumed constant for this example) = 10 m/s
- Ticks Per Revolution = 8
- Cumulative Ticks Per Second = (10 m/s)*(8 ticks/rev)/(2 m/rev) = 40
- Latency between 1PPS and measurement from wheel sensor hardware = 123 ms

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	Header	-	-	Contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary.	-	H	0
2	Latency			A measure of the latency in the velocity time tag in ms.	Ushort	2	H
3	Ticks/rev			Number of ticks per revolution	Ushort	2	H+2
4	Wheel Velocity			Short wheel velocity in ticks/s	Ushort	2	H+4
5	Reserved				Ushort	2	H+6
6	Float Wheel Velocity			Float wheel velocity in ticks/s	Float	4	H+8
7	Reserved				Ulong	4	H+12
8	Reserved				Ulong	4	H+16
9	Cumulative Ticks/s			Cumulative number of ticks/s	Ulong	4	H+20

Chapter 5 INS Logs

The INS specific logs follow the same general logging scheme as normal OEM6 Family logs. They are available in ASCII or binary formats and are defined as being either synchronous or asynchronous. All the logs in this chapter are used only with the SPAN system.

For information on other available logs and output logging, refer to the OEM6 Family Firmware Reference Manual (OM-20000129).

One difference from the standard OEM6 Family logs is there are two possible headers for the ASCII and binary versions of the logs. Which header is used for a given log is described in the log definitions in this chapter. The reason for the alternate short headers is that the normal OEM6 binary header is quite long at 28 bytes. This is nearly as long as the data portion of many of the INS logs and creates excess storage and baud rate requirements. Note that the INS related logs contain a time tag within the data block in addition to the time tag in the header. The time tag in the data block should be considered the exact time of applicability of the data. All the described INS logs except the INSCOV and INSUPDATE can be obtained at rates up to 100, 125 or 200 Hz depending on your IMU, subject to the limits of the output baud rate. The covariance and update logs are available once per second.

> Each ASCII log ends with a hexadecimal number preceded by an asterisk and followed by a line termination using the carriage return and line feed characters, for example, `*1234ABCD[CR][LF]`. This value is a 32-bit CRC of all bytes in the log, excluding the '#' or '%' identifier and the asterisk preceding the four checksum digits. See also *Description of ASCII and Binary Logs with Short Headers* on page 19.

Table 32, Inertial Solution Status on page 136 shows the status values included in the INS position, velocity and attitude output logs. If the IMU is connected properly and a good status value is not being received, check the hardware setup to ensure it is properly connected. This situation can be recognized in the RAWIMU data by observing accelerometer and gyro values which are not changing with time.

> Use a USB cable or Ethernet connection to log raw data. Serial communication is acceptable for configuring and monitoring the SPAN system through Hyperterminal or NovAtel Connect. USB or Ethernet is required if you have a post-processing application requiring 200 Hz IMU data.

5.1 Logs with INS or GNSS Data

There are several logs in the system designed to output the best available solution as well as many logs that output only a specific solution type (PSR, RTK, INS, etc). The table below lists the logs that can provide either a GNSS solution or an INS solution. Most of these derive from the solution the system picks as the best solution. SPAN systems also have a secondary best solution that derives from the GNSS solution only (BESTGNSSPOS and BESTGNSSVEL). The position output from these logs is at the phase center of the antenna.

Log	Log Format	GNSS/INS
BESTPOS	NovAtel	YES
BESTVEL	NovAtel	YES
BESTUTM	NovAtel	YES
BESTXYZ	NovAtel	YES
GPGGA	NMEA	YES
GPGLL	NMEA	YES
GPVTG	NMEA	YES

5.2 INS Specific Logs

INS specific logs provide attitude data such as roll, pitch and azimuth.

Logging Restriction Important Notice

Logging excessive amounts of high rate data can overload the system. When configuring the output for SPAN, NovAtel recommends that only one high rate (>50Hz) message be configured for output at a time. It is possible to log more than one message at high rates, but doing so could have negative impacts on the system. Also, if logging 100/125/200Hz data, always use the binary format.

For optimal performance, log only one high rate output at a time. These logs could be:

- Raw data for post processing
  RAWIMUXSB ONNEW (100, 125 or 200 Hz depending on IMU)
  - RAWIMU logs are not valid with the ONTIME trigger. The raw IMU observations contained in these logs are sequential changes in velocity and rotation. As such, you can only use them for navigation if they are logged at their full rate. See details of these logs starting on *page 181.*
- Real time INS solution
  INSPVASB ONTIME 0.01 or 0.005 (maximum rate equals the IMU rate)
  - Other possible INS solution logs available at high rates are: INSPOSSB, INSVELSB, INSATTSB

The periods available when using the ONTIME trigger are 0.005 (200Hz), 0.01 (100Hz), 0.02 (50 Hz), 0.05, 0.1, 0.2, 0.25, 0.5, 1, and any integer number of seconds.

Table 22: SPAN Logs for OEM6 - Alphabetical

ASCII Name	Description	Message ID	Type
BESTGNSSPOS	Best available GNSS position (without INS)	1429	Synch
BESTGNSSVEL	Best available GNSS velocity information (without INS)	1430	Synch
BESTLEVERARM	Distance between the IMU center of navigation and the primary GNSS antenna phase center	674	Asynch
BESTLEVERARM2	Distance between the IMU center of navigation and the secondary GNSS antenna phase center	1256	Asynch
BESTPOS	Best available combined GNSS and INS position	42	Synch
CORRIMUDATA	RAWIMU data corrected for gravity, earth's rotation and sensor errors	812	Synch
CORRIMUDATAS	RAWIMU data corrected for gravity, earth's rotation, and sensor errors (short header)	813	Synch
DELAYEDHEAVE	Log containing the value of the delayed heave filter	1709	Synch
GIMBALLEDPVA	Re-calculated gimballed position, velocity and attitude	1321	Asynch
HEAVE	Vessel heave computed by the integrated heave filter	1382	Asynch

ASCII Name	Description	Message ID	Type
IMURATECORRIMUS	RAWIMU data corrected for gravity, earth's rotation and sensor errors provided at full rate of IMU	1362	Asynch
IMURATEPVA	Most recent position, velocity and attitude at full rate of IMU	1778	Asynch
INSPVAS	Most recent position, velocity and attitude at full rate of IMU (short header)	1305	Asynch
IMUTOANTOFFSETS	Distance between the IMU center of navigation and the phase center of the GNSS antenna(s)	1270	Asynch
INSATT	Most recent attitude (roll, pitch and azimuth) measurements	263	Synch
INSATTS	Most recent attitude (roll, pitch and azimuth) measurements (short header)	319	Synch
INSATTX	Most recent attitude (roll, pitch and azimuth) measurements (roll, pitch and azimuth) with attitude standard deviation	1457	Synch
INSCOV	Position, attitude, and velocity matrices with respect to the local level frame	264	Synch
INSCOVS	Position, attitude, and velocity matrices with respect to the local level frame (short header)	320	Synch
INSPOS	Most recent position measurements	265	Synch
INSPOSS	Most recent position measurements (short header)	321	Synch
INSPOSX	Most recent position measurements with position standard deviation	1459	Synch
INSPVA	Most recent position, velocity and attitude	507	Synch
INSPVAS	Most recent position, velocity and attitude (short header)	508	Synch
INSPVAX	Most recent position, velocity and attitude with position, velocity and attitude standard deviations	1465	Synch
INSSPD	Most recent speed measurements in the horizontal and vertical directions	266	Synch
INSSPDS	Most recent speed measurements in the horizontal and vertical directions (short header)	323	Synch
INSUPDATE	Most recent INS update information	757	Asynch
INSVEL	Most recent North, East, and Up velocity vector values	267	Synch
INSVELS	Most recent North, East, and Up velocity vector values (short header)	324	Synch
INSVELX	Most recent North, East, and Up velocity vector values with velocity standard deviation	1458	Synch

ASCII Name	Description	Message ID	Type
MARK1COUNT	Count for the Mark1 input	1093	Asynch
MARK1PVA	Outputs the position, velocity and attitude when an event is received on the Mark1 input	1067	Synch
MARK2COUNT	Count for the Mark2 input	1094	Asynch
MARK2PVA	Outputs the position, velocity and attitude when an event is received on the Mark2 input	1068	Synch
MARK3COUNT	Count for the Mark3 input	1095	Asynch
MARK3PVA	Outputs the position, velocity and attitude when an event is received on the Mark3 input	1118	Synch
MARK4COUNT	Count for the Mark4 input	1096	Asynch
MARK4PVA	Outputs the position, velocity and attitude when an event is received on the Mark4 input	1119	Synch
PASHR	NMEA inertial attitude data	1177	Synch
RAWIMU	IMU status indicator and the measurements from the accelerometers and gyros	268	Asynch
RAWIMUS	IMU status indicator and the measurements from the accelerometers and gyros (short header)	325	Asynch
RAWIMUSX	IMU status indicator and the measurements from the accelerometers and gyros (extended version for post-processing with short header)	1462	Asynch
RAWIMUX	IMU status indicator and the measurements from the accelerometers and gyros (extended version for post-processing)	1461	Asynch
RELINSPVA	Relative INSPVA log	1446	Asynch
SYNCHEAVE	Synchronous log containing the instantaneous Heave value	1708	Synch
SYNCRELINSPVA	Synchronous Relative INSPVA log	1743	Synch
TAGGEDMARK1PVA	Tagged version of log that outputs the position, velocity and attitude when an event is received on the Mark1 input	1258	Synch
TAGGEDMARK2PVA	Tagged version of log that outputs the position, velocity and attitude when an event is received on the Mark2 input	1259	Synch
TAGGEDMARK3PVA	Tagged version of log that outputs the position, velocity and attitude when an event is received on the Mark3 input	1327	Synch
TAGGEDMARK4PVA	Tagged version of log that outputs the position, velocity and attitude when an event is received on the Mark4 input	1328	Synch
TIMEDWHEELDATA	Time stamped wheel sensor data	622	Asynch
TSS1	Heave, roll and pitch information in TSS1 protocol	1456	Synch

ASCII Name	Description	Message ID	Type
VARIABLELEVERARM	Displays the re-calculated variable lever arm when a new INPUTGIMBALANGLE command is received	1320	Asynch
VEHICLEBODYROTATION	Angular offset from the vehicle frame to the SPAN frame	642	
WHEELSIZE	Wheel sensor information	646	Asynch

Table 23: SPAN Logs for OEM6 - by Message ID

Message ID	ASCII Name	Description	Type
42	BESTPOS	Best available combined GNSS and INS position	Synch
263	INSATT	Most recent attitude (roll, pitch and azimuth) measurements	Synch
264	INSCOV	Position, attitude, and velocity matrices with respect to the local level frame	Synch
265	INSPOS	Most recent position measurements	Synch
266	INSSPD	Most recent speed measurements in the horizontal and vertical directions	Synch
267	INSVEL	Most recent North, East, and Up velocity vector values	Synch
268	RAWIMU	IMU status indicator and the measurements from the accelerometers and gyros	Asynch
319	INSATTS	Most recent attitude (roll, pitch and azimuth) measurements (short header)	Synch
320	INSCOVS	Position, attitude, and velocity matrices with respect to the local level frame (short header)	Synch
321	INSPOSS	Most recent position measurements (short header)	Synch
323	INSSPDS	Most recent speed measurements in the horizontal and vertical directions (short header)	Synch
324	INSVELS	Most recent North, East, and Up velocity vector values (short header)	Synch
325	RAWIMUS	IMU status indicator and the measurements from the accelerometers and gyros (short header)	Asynch
507	INSPVA	Most recent position, velocity and attitude	Synch
508	INSPVAS	Most recent position, velocity and attitude (short header)	Synch
622	TIMEDWHEELDATA	Time stamped wheel sensor data	Asynch
642	VEHICLEBODYROTATION	Angular offset from the vehicle frame to the SPAN frame	
646	WHEELSIZE	Wheel sensor information	Asynch

Message ID	ASCII Name	Description	Type
674	BESTLEVERARM	Distance between the IMU center of navigation and the primary GNSS antenna phase center	Asynch
757	INSUPDATE	Most recent INS update information	Asynch
812	CORRIMUDATA	RAWIMU data corrected for gravity, earth's rotation and sensor errors	Synch
813	CORRIMUDATAS	RAWIMU data corrected for gravity, earth's rotation and sensor errors (short header)	Synch
1067	MARK1PVA	Outputs the position, velocity and attitude when an event is received on the Mark1 input	Synch
1068	MARK2PVA	Outputs the position, velocity and attitude when an event is received on the Mark2 input	Synch
1093	MARK1COUNT	Count for the Mark1 input	Asynch
1094	MARK2COUNT	Count for the Mark2 input	Asynch
1095	MARK3COUNT	Count for the Mark3 input	Asynch
1096	MARK4COUNT	Count for the Mark4 input	Asynch
1118	MARK3PVA	Outputs the position, velocity and attitude when an event is received on the Mark3 input	Synch
1119	MARK4PVA	Outputs the position, velocity and attitude when an event is received on the Mark4 input	Synch
1177	PASHR	NMEA inertial attitude data	Synch
1256	BESTLEVERARM2	Distance between the IMU center of navigation and the secondary GNSS antenna phase center	Asynch
1258	TAGGEDMARK1PVA	Tagged version of log that outputs the position, velocity and attitude when an event is received on the Mark1 input	Synch
1259	TAGGEDMARK2PVA	Tagged version of log that outputs the position, velocity and attitude when an event is received on the Mark2 input	Synch
1270	IMUTOANTOFFSETS	Distance between the IMU center of navigation and the phase center of the GNSS antenna(s)	Asynch
1305	IMURATEPVAS	Most recent position, velocity and attitude at full rate of IMU	Asynch
1320	VARIABLELEVERARM	Displays the re-calculated variable lever arm when a new INPUTGIMBALANGLE command is received	Asynch
1321	GIMBALLEDPVA	Re-calculated gimballed position, velocity and attitude	Asynch
1327	TAGGEDMARK3PVA	Tagged version of log that outputs the position, velocity and attitude when an event is received on the Mark3 input	Synch
1328	TAGGEDMARK4PVA	Tagged version of log that outputs the position, velocity and attitude when an event is received on the Mark4 input	Synch

Message ID	ASCII Name	Description	Type
1362	IMURATECORRIMUS	RAWIMU data corrected for gravity, earth's rotation, and sensor errors provided at full rate of IMU	Asynch
1382	HEAVE	Vessel heave computed by the integrated heave filter	Asynch
1429	BESTGNSSPOS	Best available GNSS position (without INS)	Synch
1430	BESTGNSSVEL	Best available GNSS velocity information (without INS)	Synch
1446	RELINSPVA	Relative INSPVA log	Asynch
1456	TSS1	Heave, roll and pitch information in TSS1 protocol	Synch
1457	INSATTX	Most recent attitude (roll, pitch and azimuth) measurements (roll, pitch and azimuth) with attitude standard deviation	Synch
1458	INSVELX	Most recent North, East, and Up velocity vector values with velocity standard deviation	Synch
1459	INSPOSX	Most recent position measurements with position standard deviation	Synch
1461	RAWIMUX	IMU status indicator and the measurements from the accelerometers and gyros (extended version for post-processing)	Asynch
1462	RAWIMUSX	IMU status indicator and the measurements from the accelerometers and gyros (extended version for post-processing with short header)	Asynch
1465	INSPVAX	Most recent position, velocity and attitude with position, velocity and attitude standard deviations	Synch
1708	SYNCHEAVE	Synchronous log containing the instantaneous Heave value	Synch
1709	DELAYEDHEAVE	Log containing the value of the delayed heave filter	Synch
1743	SYNCRELINSPVA	Synchronous Relative INSPVA log	Synch
1778	IMURATEPVA	Most recent position, velocity and attitude at full rate of IMU	Asynch

5.2.1 BESTGNSSPOS Best GNSS Position

This log contains the best available GNSS position (without INS) computed by the receiver. In addition, it reports several status indicators, including differential age, which is useful in predicting anomalous behavior brought about by outages in differential corrections. A differential age of 0 indicates that no differential correction was used.

With the system operating in an RTK mode, this log reflects the latest low latency solution for up to 60 seconds after reception of the last base station observations. After this 60 second period, the position reverts to the best solution available and the degradation in accuracy is reflected in the standard deviation fields. If the system is not operating in an RTK mode, pseudo range differential solutions continue for the time specified in the `PSRDIFFTIMEOUT` command, refer to the *OEM6 Family Firmware Reference Manual* (OM-20000129).

> BESTGNSSPOS always outputs positions at the antenna phase center.

Message ID: 1429

Log Type: Synch

Recommended Input:

```
log bestgnssposa ontime 1
```

ASCII Example:

```
#BESTGNSSPOSA,COM1,0,92.5,FINESTEERING,1692,332119.000,00000000,8505,43521;
SOL_COMPUTED,SINGLE,51.11635530655,-114.03819448382,1064.6283,-16.9000,WGS84,1.2612,
0.9535,2.7421,"",0.000,0.000,11,11,11,11,0,06,00,03*52d3f7c0
```

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	header	Log header	-	H	0
2	Sol Status	Solution status, see *Table 24, Solution Status* on page 113	Enum	4	H
3	Pos Type	Position type, see *Table 25, Position or Velocity Type* on page 114	Enum	4	H+4
4	Lat	Latitude (degrees)	Double	8	H+8
5	Lon	Longitude (degrees)	Double	8	H+16
6	Hgt	Height above mean sea level (metres)	Double	8	H+24
7	Undulation	Undulation - the relationship between the geoid and the ellipsoid (m) of the chosen datum[a]	Float	4	H+32
8	Datum ID	Datum ID (refer *Table 29, Datum Transformation Parameters* on page 117)	Enum	4	H+36
9	Lat σ	Latitude standard deviation (metres)	Float	4	H+40
10	Lon σ	Longitude standard deviation (metres)	Float	4	H+44
11	Hgt σ	Height standard deviation (metres)	Float	4	H+48
12	Stn ID	Base station ID	Char[4]	4	H+52
13	Diff_age	Differential age in seconds	Float	4	H+56
14	Sol_age	Solution age in seconds	Float	4	H+60

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellite solutions used in solution	Uchar	1	H+65
17	#solnL1SVs	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66
18	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	Reserved		Uchar	1	H+68
20	ext sol stat	Extended solution status (see Table 88, Extended Solution Status on page 409)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 26, Galileo and BeiDou Signal-Used Mask* on page 116)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 27, GPS and GLONASS Signal-Used Mask* on page 116)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84.

Table 24: Solution Status

Binary	ASCII	Description
0	SOL_COMPUTED	Solution computed
1	INSUFFICIENT_OBS	Insufficient observations
2	NO_CONVERGENCE	No convergence
3	SINGULARITY	Singularity at parameters matrix
4	COV_TRACE	Covariance trace exceeds maximum (trace >1000 m)
5	TEST_DIST	Test distance exceeded (maximum of 3 rejections if distance >10 km)
6	COLD_START	Not yet converged from cold start
7	V_H_LIMIT	Height or velocity limits exceeded (in accordance with COCOM export licensing restrictions)
8	VARIANCE	Variance exceeds limits
9	RESIDUALS	Residuals are too large
10-12	Reserved	
13	INTEGRITY_WARNING	Solution is flagged as questionable by RAIM.
14-17	Reserved	
18	PENDING	When a `FIX POSITION` command is entered, the receiver computes its own position and determines if the fixed position is valid. [a]

Table 24: Solution Status

Binary	ASCII	Description
19	INVALID_FIX	The fixed position, entered using the `FIX POSITION` command, is not valid.
20-21	Reserved	
22	INVALID_RATE	The selected logging rate is not supported for this solution type.

a. PENDING implies not enough satellites are being tracked to verify if the FIX POSITION entered into the receiver is valid. The receiver needs to track two or more GNSS satellites to perform this check. Under normal conditions, PENDING should be seen for a few seconds on power up before the GNSS receiver locks onto the first few satellites. If the antenna is obstructed (or not plugged in) and the FIX POSITION command was entered, then PENDING may display indefinitely.

Table 25: Position or Velocity Type

Position Type (binary)	Position Type (ASCII)	Description
0	NONE	No solution
1	FIXEDPOS	Position has been fixed by the `FIX POSITION` command or by position averaging
2	FIXEDHEIGHT	Position has been fixed by the `FIX HEIGHT`, or `FIX AUTO`, command or by position averaging
3	Reserved	
4	FLOATCONV	Solution from floating point carrier phase ambiguities
5	WIDELANE	Solution from wide-lane ambiguities
6	NARROWLANE	Solution from narrow-lane ambiguities
7	Reserved	
8	DOPPLER_VELOCITY	Velocity computed using instantaneous Doppler
9-15	Reserved	
16	SINGLE	Single point position
17	PSRDIFF	Pseudorange differential solution
18	WAAS	Solution calculated using corrections from an SBAS
19	PROPOGATED	Propagated by a Kalman filter without new observations
20	OMNISTAR	OmniSTAR VBS position (L1 sub-meter) [a]
21-31	Reserved	
32	L1_FLOAT	Floating L1 ambiguity solution
33	IONOFREE_FLOAT	Floating ionospheric free ambiguity solution
34	NARROW_FLOAT	Floating narrow-lane ambiguity solution
35-47	Reserved	

Table 25: Position or Velocity Type (continued)

Position Type (binary)	Position Type (ASCII)	Description
48	L1_INT	Integer L1 ambiguity solution
49	WIDE_INT	Integer wide-lane ambiguity solution
50	NARROW_INT	Integer narrow-lane ambiguity solution
51	RTK_DIRECT_INS	RTK status where the RTK filter is directly initialized from the INS filter[b]
52	INS_SBAS	INS calculated position corrected for the antenna [b]
53	INS_PSRSP	INS pseudorange single point solution – no DGPS corrections [b]
54	INS_PSRDIFF	INS pseudorange differential solution [b]
55	INS_RTKFLOAT	INS RTK floating point ambiguities solution [b]
56	INS_RTKFIXED	INS RTK fixed ambiguities solution [b]
57	INS_OMNISTAR	INS OmniSTAR VBS position (L1 sub-meter) [a]
58	INS_OMNISTAR_HP	INS OmniSTAR high precision solution [a]
59	INS_OMNISTAR_XP	INS OmniSTAR extra precision solution [a]
60-63	Reserved	
64	OMNISTAR_HP	OmniSTAR high precision [a]
65	OMNISTAR_XP	OmniSTAR extra precision [a]
66-67	Reserved	
68	PPP_CONVERGING	Precise Point Positioning (PPP) solution converging
69	PPP	Precise Point Positioning (PPP) solution
70-72	Reserved	
73	INS_PPP_Converging	INS NovAtel CORRECT Precise Point Positioning (PPP) solution converging
74	INS_PPP	INS NovAtel CORRECT Precise Point Positioning (PPP) solution

a. In addition to a NovAtel receiver with L-Band capability, a subscription to the OmniSTAR service is required. Contact NovAtel for details.

b. These types appear in position logs such as BESTPOS.

Table 26: Galileo and BeiDou Signal-Used Mask

Bit	Mask	Description
0	0x01	Galileo E1 used in Solution
1-3	0x02 - 0x08	Reserved
4	0x10	BeiDou B1 used in Solution
5	0x20	BeiDou B2 used in Solution
6-7	0x40-0x80	Reserved

Table 27: GPS and GLONASS Signal-Used Mask

Bit	Mask	Description
0	0x01	GPS L1 used in Solution
1	0x02	GPS L2 used in Solution
2	0x04	GPS L5 used in Solution
3	0x08	Reserved
4	0x10	GLONASS L1 used in Solution
5	0x20	GLONASS L2 used in Solution
6-7	0x40-0x80	Reserved

Table 28: Extended Solution Status

Bit	Mask	Description
0	0x01	AdVance RTK Verified 0: Not Verified 1: Verified
1-3	0x0E	Pseudorange Iono Correction 0: Unknown[a] 1: Klobuchar Broadcast 2: SBAS Broadcast 3: Multi-frequency Computed 4: PSRDiff Correction 5: NovAtel Blended Iono value
4-7	0xF0	Reserved

a. Unknown can indicate that the Iono Correction type is None or that the default Klobuchar parameters are being used.

Table 29: Datum Transformation Parameters

Datum ID# [a]	NAME	DX [b]	DY [b]	DZ [b]	DATUM DESCRIPTION	ELLIPSOID
1	ADIND	-162	-12	206	This datum has been updated, see ID# 65 [c]	Clarke 1880
2	ARC50	-143	-90	-294	ARC 1950 (SW & SE Africa)	Clarke 1880
3	ARC60	-160	-8	-300	This datum has been updated, see ID# 66 [c]	Clarke 1880
4	AGD66	-133	-48	148	Australian Geodetic Datum 1966	Australian National
5	AGD84	-134	-48	149	Australian Geodetic Datum 1984	Australian National
6	BUKIT	-384	664	-48	Bukit Rimpah (Indonesia)	Bessel 1841
7	ASTRO	-104	-129	239	Camp Area Astro (Antarctica)	International 1924
8	CHATM	175	-38	113	Chatham 1971 (New Zealand)	International 1924
9	CARTH	-263	6	431	Carthage (Tunisia)	Clarke 1880
10	CAPE	-136	-108	-292	CAPE (South Africa)	Clarke 1880
11	DJAKA	-377	681	-50	Djakarta (Indonesia)	Bessel 1841
12	EGYPT	-130	110	-13	Old Egyptian	Helmert 1906
13	ED50	-87	-98	-121	European 1950	International 1924
14	ED79	-86	-98	-119	European 1979	International 1924
15	GUNSG	-403	684	41	G. Segara (Kalimantan - Indonesia)	Bessel 1841
16	GEO49	84	-22	209	Geodetic Datum 1949 (New Zealand)	International 1924
17	GRB36	375	-111	431	Do not use. Use ID# 76 instead. [d]	Airy 1830
18	GUAM	-100	-248	259	Guam 1963 (Guam Island)	Clarke 1866
19	HAWAII	89	-279	-183	Do not use. Use ID# 77 or ID# 81 instead. [d]	Clarke 1866
20	KAUAI	45	-290	-172	Do not use. Use ID# 78 or ID# 82 instead. [d]	Clarke 1866
21	MAUI	65	-290	-190	Do not use. Use ID# 79 or ID# 83 instead. [d]	Clarke 1866
22	OAHU	56	-284	-181	Do not use. Use ID# 80 or ID# 84 instead. [d]	Clarke 1866
23	HERAT	-333	-222	114	Herat North (Afghanistan)	International 1924
24	HJORS	-73	46	-86	Hjorsey 1955 (Iceland)	International 1924
25	HONGK	-156	-271	-189	Hong Kong 1963	International 1924
26	HUTZU	-634	-549	-201	This datum has been updated, see ID# 68 [c]	International 1924
27	INDIA	289	734	257	Do not use. Use ID# 69 or ID# 70 instead. [d]	Everest (EA)
28	IRE65	506	-122	611	Do not use. Use ID# 71 instead. [d]	Modified Airy
29	KERTA	-11	851	5	Kertau 1948 (West Malaysia and Singapore)	Everest (EE)
30	KANDA	-97	787	86	Kandawala (Sri Lanka)	Everest (EA)
31	LIBER	-90	40	88	Liberia 1964	Clarke 1880

Table 29: Datum Transformation Parameters (continued)

Datum ID# [a]	NAME	DX [b]	DY [b]	DZ [b]	DATUM DESCRIPTION	ELLIPSOID
32	LUZON	-133	-77	-51	Do not use. Use ID# 72 instead. [d]	Clarke 1866
33	MINDA	-133	-70	-72	This datum has been updated, see ID# 73 [c]	Clarke 1866
34	MERCH	31	146	47	Merchich (Morocco)	Clarke 1880
35	NAHR	-231	-196	482	This datum has been updated, see ID# 74 [c]	Clarke 1880
36	NAD83	0	0	0	N. American 1983 (Includes Areas 37-42)	GRS-80
37	CANADA	-10	158	187	N. American Canada 1927	Clarke 1866
38	ALASKA	-5	135	172	N. American Alaska 1927	Clarke 1866
39	NAD27	-8	160	176	N. American Conus 1927	Clarke 1866
40	CARIBB	-7	152	178	This datum has been updated, see ID# 75 [c]	Clarke 1866
41	MEXICO	-12	130	190	N. American Mexico	Clarke 1866
42	CAMER	0	125	194	N. American Central America	Clarke 1866
43	MINNA	-92	-93	122	Nigeria (Minna)	Clarke 1880
44	OMAN	-346	-1	224	Oman	Clarke 1880
45	PUERTO	11	72	-101	Puerto Rica and Virgin Islands	Clarke 1866
46	QORNO	164	138	-189	Qornoq (South Greenland)	International 1924
47	ROME	-255	-65	9	Rome 1940 Sardinia Island	International 1924
48	CHUA	-134	229	-29	South American Chua Astro (Paraguay)	International 1924
49	SAM56	-288	175	-376	South American (Provisional 1956)	International 1924
50	SAM69	-57	1	-41	South American 1969	S. American 1969
51	CAMPO	-148	136	90	S. American Campo Inchauspe (Argentina)	International 1924
52	SACOR	-206	172	-6	South American Corrego Alegre (Brazil)	International 1924
53	YACAR	-155	171	37	South American Yacare (Uruguay)	International 1924
54	TANAN	-189	-242	-91	Tananarive Observatory 1925 (Madagascar)	International 1924
55	TIMBA	-689	691	-46	This datum has been updated, see ID# 85 [c]	Everest (EB)
56	TOKYO	-128	481	664	This datum has been updated, see ID# 86 [c]	Bessel 1841
57	TRIST	-632	438	-609	Tristan Astro 1968 (Tristan du Cunha)	International 1924
58	VITI	51	391	-36	Viti Levu 1916 (Fiji Islands)	Clarke 1880
59	WAK60	101	52	-39	This datum has been updated, see ID# 67 [c]	Hough 1960
60	WGS72	0	0	4.5	World Geodetic System - 72	WGS72
61	WGS84	0	0	0	World Geodetic System - 84	WGS84
62	ZANDE	-265	120	-358	Zanderidj (Surinam)	International 1924

Table 29: Datum Transformation Parameters (continued)

Datum ID# [a]	NAME	DX [b]	DY [b]	DZ [b]	DATUM DESCRIPTION	ELLIPSOID
63	USER	0	0	0	User Defined Datum Defaults	User [a]
64	CSRS	Time-variable 7 parameter transformation				
65	ADIM	-166	-15	204	Adindan (Ethiopia, Mali, Senegal & Sudan) [c]	Clarke 1880
66	ARSM	-160	-6	-302	ARC 1960 (Kenya, Tanzania) [c]	Clarke 1880
67	ENW	102	52	-38	Wake-Eniwetok (Marshall Islands) [c]	Hough 1960
68	HTN	-637	-549	-203	Hu-Tzu-Shan (Taiwan) [c]	International 1924
69	INDB	282	726	254	Indian (Bangladesh) [d]	Everest (EA)
70	INDI	295	736	257	Indian (India, Nepal) [d]	Everest (EA)
71	IRL	506	-122	611	Ireland 1965 [d]	Modified Airy
72	LUZA	-133	-77	-51	Luzon (Philippines excluding Mindanoa Is.) [de]	Clarke 1866
73	LUZB	-133	-79	-72	Mindanoa Island [c]	Clarke 1866
74	NAHC	-243	-192	477	Nahrwan (Saudi Arabia) [c]	Clarke 1880
75	NASP	-3	142	183	N. American Caribbean [c]	Clarke 1866
76	OGBM	375	-111	431	Great Britain 1936 (Ordinance Survey) [d]	Airy 1830
77	OHAA	89	-279	-183	Hawaiian Hawaii [d]	Clarke 1866
78	OHAB	45	-290	-172	Hawaiian Kauai [d]	Clarke 1866
79	OHAC	65	-290	-190	Hawaiian Maui [d]	Clarke 1866
80	OHAD	58	-283	-182	Hawaiian Oahu [d]	Clarke 1866
81	OHIA	229	-222	-348	Hawaiian Hawaii [d]	International 1924
82	OHIB	185	-233	-337	Hawaiian Kauai [d]	International 1924
83	OHIC	205	-233	-355	Hawaiian Maui [d]	International 1924
84	OHID	198	-226	-347	Hawaiian Oahu [d]	International 1924
85	TIL	-679	669	-48	Timbalai (Brunei and East Malaysia) 1948 [c]	Everest (EB)
86	TOYM	-148	507	685	Tokyo (Japan, Korea and Okinawa) [c]	Bessel 1841

a. The default user datum is WGS84. See also the USERDATUM and USEREXPDATUM commands. The following logs report the datum used according to the OEM card Datum ID column: BESTPOS, BESTUTM, MATCHEDPOS and PSRPOS. Descriptions of these commands and logs are available in the *OEM6 Family Firmware Reference Manual* (OM-20000129).

b. The DX, DY and DZ offsets are from your local datum to WGS84.

c. The updated datum have the new x, y and z translation values updated to the latest numbers. The old datum values can still be used for backwards compatibility.

d. Use the corrected datum only (with the higher ID#) as the old datum is incorrect.

e. The original LUZON values are the same as for LUZA but the original has an error in the code.

5.2.2 BESTGNSSVEL
Best Available GNSS Velocity Data

This log contains the best available GNSS velocity information (without INS) computed by the receiver. In addition, it reports a velocity status indicator, which is useful to indicate whether or not the corresponding data is valid. The velocity measurements sometimes have a latency associated with them. The time of validity is the time tag in the log minus the latency value.

The velocity is typically computed from the average change in pseudorange over the time interval or the RTK Low Latency filter. As such, it is an average velocity based on the time difference between successive position computations and not an instantaneous velocity at the BESTGNSSVEL time tag. The velocity latency to be subtracted from the time tag is normally half the time between filter updates. Under default operation, the positioning filters are updated at a rate of 2 Hz. This translates into a velocity latency of 0.25 seconds. The latency is reduced by increasing the update rate of the positioning filter used by requesting the BESTGNSSVEL or BESTGNSSPOS messages at a rate higher than 2 Hz. For example, a logging rate of 10 Hz reduces the velocity latency to 0.005 seconds. For integration purposes, the velocity latency should be applied to the record time tag.

A valid solution with a latency of 0.0 indicates the instantaneous Doppler measurement was used to calculate velocity.

Message ID: 1430

Log Type: Synch

Recommended Input:

```
log bestgnssvela ontime 1
```

ASCII Example:

```
#BESTGNSSVELA,COM1,0,91.5,FINESTEERING,1692,332217.000,00000000,00b0,43521;
SOL_COMPUTED,DOPPLER_VELOCITY,0.150,0.000,0.0168,323.193320,0.0232,0.0*159c13ad
```

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	header	Log header	-	H	0
2	Sol Status	Solution status, see *Table 24, Solution Status* on page 113	Enum	4	H
3	Vel Type	Velocity type, see *Table 25, Position or Velocity Type* on page 114	Enum	4	H+4
4	Latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results.	Float	4	H+8
5	Age	Differential age	Float	4	H+12
6	Hor Spd	Horizontal speed over ground, in metres per second	Double	8	H+16
7	Trk Gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees	Double	8	H+24
8	Vert Spd	Vertical speed, in metres per second, where positive values indicate increasing altitude (up) and negative values indicate decreasing altitude (down)	Double	8	H+32
9	Reserved		Float	4	H+40
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.3 BESTLEVERARM IMU to Antenna Lever Arm

The BESTLEVERARM log contains the distance between the IMU center of navigation and the primary GNSS antenna phase center in the IMU enclosure frame and its associated uncertainties. If the lever arm was entered using the `SETIMUTOANTOFFSET` command, see *SETIMUTOANTOFFSET* on page 79, these values are reflected in this log. When the lever arm calibration is complete, see *LEVERARMCALIBRATE* on page 63, the solved values are also output in this log.

The values in the BESTLEVERARM log is also available in the IMUTOANTOFFSETS log, see page 133.

The default X (pitch), Y (roll) and Z (azimuth) directions of the IMU enclosure frame are clearly marked on the IMU, see *Figure 2, Frame of Reference* on page 75.

Message ID: **674**

Log Type: **Asynch**

Recommended Input:

```
log bestleverarma onchanged
```

ASCII Example:

```
#BESTLEVERARMA,COM1,0,83.5,UNKNOWN,0,2.983,00000008,39e4,35484;0.3934000000000000,
-1.2995000000000001,0.0105500000000000,0.0300000000000000,0.0300000000000000,
0.0300000000000000,4*876c47ad
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	X Offset	IMU Enclosure Frame (m)	Double	8	H
3	Y Offset	IMU Enclosure Frame (m)	Double	8	H+8
4	Z Offset	IMU Enclosure Frame (m)	Double	8	H+16
5	X Uncertainty	IMU Enclosure Frame (m)	Double	8	H+24
6	Y Uncertainty	IMU Enclosure Frame (m)	Double	8	H+32
7	Z Uncertainty	IMU Enclosure Frame (m)	Double	8	H+40
8	iMapping	See *Table 21, Full Mapping Definitions* on page 76	Integer	4	H+48
9	xxxx	32-bit CRC	Hex	4	H+52
10	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.4 BESTLEVERARM2 IMU to Antenna Lever Arm

The BESTLEVERARM2 log contains the distance between the IMU center of navigation and the secondary GNSS antenna phase center in the IMU enclosure frame. The second lever arm cannot be calibrated so must be entered using the SETIMUTOANTOFFSET2 command, see *SETIMUTOANTOFFSET2* on page 81.

The values in the BESTLEVERARM2 log is also available in the IMUTOANTOFFSETS log, see page 133.

The default X (pitch), Y (roll) and Z (azimuth) directions of the IMU enclosure frame are clearly marked on the IMU, see *Figure 2, Frame of Reference* on page 75.

Message ID: 1256

Log Type: Asynch

Recommended Input:

```
log bestleverarm2a onchanged
```

ASCII Example:

```
#BESTLEVERARM2A,COM1,0,83.5,UNKNOWN,0,2.983,00000008,39e4,35484;0.3934000000000000,
-1.2995000000000001,0.0105500000000000,0.0300000000000000,0.0300000000000000,
0.0300000000000000,4*876c47ad
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	X Offset	IMU Enclosure Frame (m)	Double	8	H
3	Y Offset	IMU Enclosure Frame (m)	Double	8	H+8
4	Z Offset	IMU Enclosure Frame (m)	Double	8	H+16
5	X Uncertainty	IMU Enclosure Frame (m)	Double	8	H+24
6	Y Uncertainty	IMU Enclosure Frame (m)	Double	8	H+32
7	Z Uncertainty	IMU Enclosure Frame (m)	Double	8	H+40
8	iMapping	See *Table 21, Full Mapping Definitions* on page 76	Integer	4	H+48
9	xxxx	32-bit CRC	Hex	4	H+52
10	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.5 BESTPOS
Best Position

This log contains the best available combined GNSS and Inertial Navigation System (INS - if available) position (in metres) computed by the receiver. In addition, it reports several status indicators, including differential age, which is useful in predicting anomalous behavior brought about by outages in differential corrections. A differential age of 0 indicates that no differential correction was used.

With the system operating in an RTK mode, this log reflects the latest low-latency solution for up to 60 seconds after reception of the last base station observations. After this 60 second period, the position reverts to the best solution available; the degradation in accuracy is reflected in the standard deviation fields. If the system is not operating in an RTK mode, pseudo range differential solutions continue for the time specified in the `PSRDIFFTIMEOUT` command, refer to the *OEM6 Family Firmware Reference Manual* (OM-20000129).

> BESTPOS always outputs positions at the antenna phase center.

Message ID: 42

Log Type: Synch

Recommended Input:

```
log bestposa ontime 1
```

ASCII Example 1:

```
#BESTPOSA,COM1,0,83.5,FINESTEERING,1419,336148.000,00000040,6145,2724;SOL_COMPUTED,
SINGLE,51.11636418888,-114.03832502118,1064.9520,-16.2712,WGS84,1.6961,1.3636,
3.6449,"",0.000,0.000,8,8,0,0,0,06,0,03*6f63a93d
```

ASCII Example 2:

```
#BESTPOSA,COM1,0,78.5,FINESTEERING,1419,336208.000,00000040,6145,2724;SOL_COMPUTED,
NARROW_INT,51.11635910984,-114.03833105168,1063.8416,-16.2712, WGS84,0.0135,0.0084,
0.0172,"AAAA",1.000,0.000,8,8,8,8,0,01,0,03*3d9fbd48
```

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	BESTPOS header	Log header		H	0
2	sol stat	Solution status, see *Table 24, Solution Status* on page 113	Enum	4	H
3	pos type	Position type, see *Table 25, Position or Velocity Type* on page 114	Enum	4	H+4
4	lat	Latitude (degrees)	Double	8	H+8
5	lon	Longitude (degrees)	Double	8	H+16
6	hgt	Height above mean sea level (metres)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the ellipsoid (m) of the chosen datum [a]	Float	4	H+32
8	datum id#	Datum ID number, see *Table 29, Datum Transformation Parameters* on page 117	Enum	4	H+36
9	lat σ	Latitude standard deviation (metres)	Float	4	H+40

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
10	lon σ	Longitude standard deviation (metres)	Float	4	H+44
11	hgt σ	Height standard deviation (metres)	Float	4	H+48
12	stn id	Base station ID	Char[4]	4	H+52
13	diff_age	Differential age in seconds	Float	4	H+56
14	sol_age	Solution age in seconds	Float	4	H+60
15	#SVs	Number of satellite vehicles tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65
17	#solnL1SVs	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66
18	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	Reserved		Uchar	1	H+68
20	ext sol stat	Extended solution status (see *Table 28, Extended Solution Status* on page 116)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 26, Galileo and BeiDou Signal-Used Mask* on page 116)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 27, GPS and GLONASS Signal-Used Mask* on page 116)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84

5.2.6 CORRIMUDATA
Corrected IMU Measurements

The CORRIMUDATA log contains the RAWIMU data corrected for gravity, the earth's rotation and estimated sensor errors. The values in this log are instantaneous incremental values, in units of radians for the attitude rate and m/s for the accelerations. To get the full attitude rate and acceleration values, multiply the values in the CORRIMUDATA log by the data rate of the IMU in Hz.

> The short header format, CORRIMUDATAS, is recommended, as it is for all high data rate logs.

CORRIMUDATA can be logged with the ONTIME trigger, up to the full data rate of the IMU.

> Since the CORRIMUDATA values are instantaneous, if you log at a rate less than full data rate of the IMU, the corrected IMU data is received at the epoch closest to the requested time interval.
>
> For asynchronous, full rate data, see *IMURATECORRIMUS* on page 130.

If the IMU is mounted with the z-axis pointed up, as marked on the enclosure, the SPAN computation frame is the same as the IMU enclosure frame. The x, y, and z axes referenced in this log are of the SPAN computational frame by default. For more information on how the SPAN computational frame relates to the IMU enclosure frame, see the relevant SPAN User Manual and the `SETIMUORIENTATION` command on page 74. If the `APPLYVEHICLEBODYROTATION` command has been enabled (see page 36), the values in CORRIMUDATA log are in the vehicle frame, not the SPAN computation frame.

Message ID: **812**

Log Type: **Synch**

Recommended Input:

```
log corrimudatab ontime 0.01
```

Example log:

```
#CORRIMUDATAA,COM1,0,77.5,FINESTEERING,1769,237601.000,00000020,bdba,12597;1769,
237601.000000000,-0.000003356,0.000002872,0.000001398,0.000151593,0.000038348,
-0.000078820*1f7eb709
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS week	Ulong	4	H+
3	Seconds	GNSS seconds from week start	Double	8	H+4
4	PitchRate	About x axis rotation (right-handed) (rad/s/sample)	Double	8	H+12
5	RollRate	About y axis rotation (right-handed) (rad/s/sample)	Double	8	H+20
6	YawRate	About z axis rotation (right-handed) (rad/s/sample)	Double	8	H+28
7	LateralAcc	INS Lateral Acceleration (along x axis) (m/s/sample)	Double	8	H+36
8	LongitudinalAcc	INS Longitudinal Acceleration (along y axis) (m/s/sample)	Double	8	H+44
9	VerticalAcc	INS Vertical Acceleration (along z axis) (m/s/sample)	Double	8	H+52
10	xxxx	32-bit CRC	Hex	4	H+56
11	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.7 CORRIMUDATAS Short Corrected IMU Measurements

The CORRIMUDATAS log is the short header version of the CORRIMUDATA log (page 125) This log contains the RAWIMU data corrected for gravity, the earth's rotation and estimated sensor errors. The values in this log are instantaneous, incremental values, in units of radians for the attitude rate and m/s for the accelerations. To get the full attitude rate and acceleration values, multiply the values in the CORRIMUDATAS log by the data rate of your IMU in Hz.

CORRIMUDATAS can be logged with the ONTIME trigger, up to the full data rate of the IMU.

> Since the CORRIMUDATA values are instantaneous, if you log at a rate less than full data rate of the IMU, the corrected IMU data is received at the epoch closest to the requested time interval.
>
> For asynchronous, full rate data, see *IMURATECORRIMUS* on page 130.

If the IMU is mounted with the z-axis pointed up, as marked on the enclosure, the SPAN computation frame is the same as the IMU enclosure frame. The x, y, and z axes referenced in this log are of the SPAN computational frame by default. For more information on how the SPAN computational frame relates to the IMU enclosure frame, see the relevant SPAN User Manual and the `SETIMUORIENTATION` command on page 74. If the `APPLYVEHICLEBODYROTATION` command has been enabled (see page 36), the values in CORRIMUDATAS log are in the vehicle frame, not the SPAN computation frame.

Message ID: 813

Log Type: Synch

Recommended Input:

```
log corrimudatasb ontime 0.01
```

Example log:

```
%CORRIMUDATASA,1581,341553.000;1581,341552.997500000,-0.000000690,-0.000001549,
0.000001654,0.000061579,-0.000012645,-0.000029988*770c6232
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS week	Ulong	4	H+
3	Seconds	GNSS seconds from week start	Double	8	H+4
4	PitchRate	About x-axis rotation (right-handed) (rad/s/sample)	Double	8	H+12
5	RollRate	About y-axis rotation (right-handed) (rad/s/sample)	Double	8	H+20
6	YawRate	About z-axis rotation (right-handed) (rad/s/sample)	Double	8	H+28
7	LateralAcc	INS Lateral Acceleration (along x-axis) (m/s/sample)	Double	8	H+36
8	LongitudinalAcc	INS Longitudinal Acceleration (along y-axis) (m/s/sample)	Double	8	H+44
9	VerticalAcc	INS Vertical Acceleration (along z-axis) (m/s/sample)	Double	8	H+52
10	xxxx	32-bit CRC	Hex	4	H+56
11	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.8 DELAYEDHEAVE
Delayed Heave Filter

This log contains the value of the delayed heave filter. The delayed heave value differs from the heave value in that delayed heave uses forward and backward smoothing, while heave uses backward smoothing only.

The heave filter must be enabled using the `HEAVEFILTER` command, see page 52, before this log is available.

Message ID: 1709

Log Type: Synch

Recommended Input:

```
log delayedheavea ontime 0.1
```

ASCII example:

```
#DELAYEDHEAVEA,COM1,0,72.0,FINESTEERING,1769,237598.000,00000020,27a3,12597;
0.000080643,0.086274510*85cdb46d
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	Delayed Heave	Delayed heave value	Double	8	H
3	Std. Dev.	Standard deviation of the delayed heave value	Double	8	H+8
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+16
5	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.9 GIMBALLEDPVA
Display Gimballed Position

Use the GIMBALLEDPVA log to view the re-calculated gimballed position, velocity and attitude whenever a new INPUTGIMBALANGLE command is received.

The log is not output until the INS alignment is complete.

Message ID: 1321

Log Type: Asynch

Recommended Input:

```
log gimballedpvaa onnew
```

ASCII Example:

```
#GIMBALLEDPVAA,COM1,0,93.5,FINESTEERING,1635,320568.514,00000000,0000,407;1635,
320568.514000000,51.116376614,-114.038259915,1046.112025828,-0.000291756,-0.000578067,
0.030324466,-0.243093917,-0.127718304,19.495023227, INS_ALIGNMENT_COMPLETE*32fbb61b
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	Week	GPS week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	WGS84 latitude in degrees	Double	8	H+12
5	Longitude	WGS84 longitude in degrees	Double	8	H+20
6	Height	WGS84 ellipsoidal height	Double	8	H+28
7	North Velocity	Velocity in a northerly direction	Double	8	H+36
8	East Velocity	Velocity in an easterly direction	Double	8	H+44
9	Up Velocity	Velocity in an upward direction	Double	8	H+52
10	Roll	Right-handed rotation from local level around the y-axis in degrees	Double	8	H+60
11	Pitch	Right-handed rotation from local level around the x-axis in degrees	Double	8	H+68
12	Azimuth	Right-handed rotation from local level around the z-axis in degrees	Double	8	H+76
13	Status	INS status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.10 HEAVE
Heave Filter Log

This log provides vessel heave computed by the integrated heave filter. Refer also to information in *SETHEAVEWINDOW* on page 73. This log is asynchronous, but is available at approximately 10 Hz.

You must have an inertial solution to use this log.

The heave filter must be enabled using the `HEAVEFILTER` command, see page 52, before this log is available.

Message ID: 1382

Log Type: Asynch

Recommended Input:

```
log heavea onnew
```

Example:

```
#HEAVEA,USB1,0,38.5,FINESTEERING,1630,232064.599,00000000,a759,6696;1630,232064.58988539
2,0.086825199*93392cb4
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Heave	Instantaneous heave in metres	Double	8	H+12
5	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+20
6	[CR][LF]	Sentence Terminator (ASCII Only)	-	-	-

5.2.11 IMURATECORRIMUS Asynchronous Corrected IMU Data

This log provides the same information as the CORRIMUDATA log, but is available asynchronously at the full rate of the IMU.

> Using this log consumes significant system resources and should only be used by experienced users.

To use this log, asynchronous logging must be enabled. See *ASYNCHINSLOGGING* on page 37.

Message ID: 1362

Log Type: Asynch

Recommended Input:

```
log imuratecorrimus
```

Example log:

```
%IMURATECORRIMUSA,1581,341553.000;1581,341552.997500000,-0.000000690,-0.000001549,
0.000001654,0.000061579,-0.000012645,-0.000029988*770c6232
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS week	Ulong	4	H+
3	Seconds	GNSS seconds from week start	Double	8	H+4
4	PitchRate	About x axis rotation (rad/s/sample)	Double	8	H+12
5	RollRate	About y axis rotation (rad/s/sample)	Double	8	H+20
6	YawRate	About z axis rotation (right-handed) (rad/s/sample)	Double	8	H+28
7	LateralAcc	INS Lateral Acceleration (along x-axis) (m/s/sample)	Double	8	H+36
8	LongitudinalAcc	INS Longitudinal Acceleration (along y-axis) (m/s/sample)	Double	8	H+44
9	VerticalAcc	INS Vertical Acceleration (along z-axis) (m/s/sample)	Double	8	H+52
10	xxxx	32-bit CRC	Hex	4	H+56
11	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.12 IMURATEPVA
Asynchronous INS Position, Velocity and Attitude

This log provides the same information as the INSPVA log, but is available asynchronously at the full rate of the IMU.

> Using this log consumes significant system resources and should only be used by experienced users.
>
> However, using this log consumes less resources than logging the synchronous INSPVA log at the same rate.

To use this log, asynchronous logging must be enabled. See *ASYNCHINSLOGGING* on page 37.

Message ID: **1778**

Log Type: **Asynch**

Recommended Input:

```
log imuratepvaa onnew
```

ASCII Example:

```
#IMURATEPVAA,COM1,0,57.0,FINESTEERING,1802,320345.180,00000000,9b1f,12987;1802,320345.18
0000030,51.11695246671,-114.03897779953,1047.6905,-0.2284,0.0076,0.2227,0.160588332,-
0.039823409,269.988184416,INS_ALIGNMENT_COMPLETE*f60016a6
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84) [degrees]	Double	8	H+12
5	Longitude	Longitude (WGS84) [degrees]	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84) [m]	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction) [m/s]	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction) [m/s]	Double	8	H+44
9	Up Velocity	Velocity in an up direction [m/s]	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+76
13	Status	INS Status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.13 IMURATEPVAS Asynchronous INS Position, Velocity and Attitude

This log provides the same information as the INSPVAS log, but is available asynchronously at the full rate of the IMU.

> Using this log consumes significant system resources and should only be used by experienced users.

To use this log, asynchronous logging must be enabled. See *ASYNCHINSLOGGING* on page 37.

Message ID: 1305

Log Type: Asynch

Recommended Input:

```
log imuratepvas
```

ASCII Example:

```
%IMURATEPVASA,1264,144059.000;1264,144059.002135700,51.116680071,-114.037929194,
515.286704183,277.896368884,84.915188605,-8.488207941,0.759619515,-2.892414901,
6.179554750,INS_ALIGNMENT_COMPLETE*855d6f76
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84)	Double	8	H+12
5	Longitude	Longitude (WGS84)	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84) [m]	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction) [m/s]	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction) [m/s]	Double	8	H+44
9	Up Velocity	Velocity in an up direction [m/s]	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North	Double	8	H+76
13	Status	INS Status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.14 IMUTOANTOFFSETS IMU to Antenna(s) Lever Arm

This log contains the distance between the IMU and the GNSS antenna(s) in the IMU enclosure frame and its associated uncertainties. This log contains the same information as the BESTLEVERARM logs for each lever arm, but is intended as a single source for all lever arm information available on the system.

Message ID: 1270

Log Type: Asynch

Recommended Input:

```
log imutoantoffsetsa onchanged
```

ASCII Example:

```
#IMUTOANTOFFSETSA,COM1,0,98.5,FINESTEERING,1581,339209.733,60000041,0000,265;0,1,
LEVER_ARM_PRIMARY,-0.326000000,0.126000000,1.285000000,0.032600000,0.012600000,
0.128500000,LEVER_ARM_FROM_COMMAND*8f0f90b5
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Header	Log Header	-	H	0
2	IMU Orientation	See *Table 21, Full Mapping Definitions* on page 76	Long	4	H
3	Number of Entries	Number of stored lever arms	Long	4	H+4
4	Lever Arm Type	Type of lever arm. See *Table 30, Lever Arm Type* on page 134.	Enum	4	H+8
5	X Offset	IMU Enclosure Frame (m)	Double	8	H+12
6	Y Offset	IMU Enclosure Frame (m)	Double	8	H+20
7	Z Offset	IMU Enclosure Frame (m)	Double	8	H+28
8	X Uncertainty	IMU Enclosure Frame (m)	Double	8	H+36
9	Y Uncertainty	IMU Enclosure Frame (m)	Double	8	H+44
10	Z Uncertainty	IMU Enclosure Frame (m)	Double	8	H+52
11	Lever Arm Source	Source of the lever arm. See *Table 31, Lever Arm Source* on page 134 for the different values	Enum	4	H+60
12…	Next component offset = H + 8 + (#comp * 56)				
variable	XXXX	32-bit CRC (ASCII and Binary only)	Hex	4	H+8+ (#comp*56)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 30: Lever Arm Type

Value (binary)	Lever Arm Source (ASCII)	Description
0	LEVER_ARM_INVALID	An invalid lever arm.
1	LEVER_ARM_PRIMARY	Primary lever arm entered for all SPAN systems.
2	LEVER_ARM_SECONDARY	Secondary lever arm entered for dual-antenna SPAN systems.

Table 31: Lever Arm Source

Value (binary)	Lever Arm Source (ASCII)	Description
0	LEVER_ARM_NONE	No lever arm exists.
1	LEVER_ARM_FROM_NVM	Lever arm restored from NVM.
2	LEVER_ARM_CALIBRATING	Lever arm currently calibrating.
3	LEVER_ARM_CALIBRATED	Lever arm computed from calibration routine.
4	LEVER_ARM_FROM_COMMAND	Lever arm entered via command.

5.2.15 INSATT INS Attitude

This log contains the most recent attitude measurements corresponding to the SPAN frame axis according to how the IMU was installed and configured. The attitude measurements may not correspond to other definitions of the terms pitch, roll and azimuth. If the IMU z-axis (as marked on the enclosure) is not pointing up, the output attitude is with respect to the SPAN computational frame, and not the frame marked on the enclosure. See *SETIMUORIENTATION* on page 74 to determine what the SPAN computation frame will be, given how your IMU is mounted. To output the attitude in the vehicle frame, see page 36 for information about the `APPLYVEHICLEBODYROTATION` command.

Message ID: 263

Log Type: Synch

Recommended Input:

```
log insatta ontime 1
```

ASCII Example:

```
#INSATTA,USB2,0,14.5,FINESTEERING,1541,487970.000,00040000,5b35,37343;1541,487970.000549
050,1.876133508,-4.053672765,328.401460897,INS_SOLUTION_GOOD*ce4ac533
```

Field	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Roll	Right-handed rotation from local level around y-axis in degrees.	Double	8	H+12
5	Pitch	Right-handed rotation from local level around x-axis in degrees.	Double	8	H+20
6	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North. This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+28
7	Status	INS status, see *Table 32, Inertial Solution Status* on page 136.	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 32: Inertial Solution Status

Binary	ASCII	Description
0	INS_INACTIVE	IMU logs are present, but the alignment routine has not started; INS is inactive.
1	INS_ALIGNING	INS is in alignment mode.
2	INS_HIGH_VARIANCE	The INS solution is in navigation mode but the azimuth solution uncertainty has exceeded the threshold. The default threshold is 2 degrees for most IMUs.[a] The solution is still valid but you should monitor the solution uncertainty in the INSCOV log. You may encounter this state during times when the GNSS, used to aid the INS, is absent.
3	INS_SOLUTION_GOOD	The INS filter is in navigation mode and the INS solution is good.
6	INS_SOLUTION_FREE	The INS filter is in navigation mode and the GNSS solution is suspected to be in error. This may be due to multipath or limited satellite visibility. The inertial filter has rejected the GNSS position and is waiting for the solution quality to improve.
7	INS_ALIGNMENT_COMPLETE	The INS filter is in navigation mode, but not enough vehicle dynamics have been experienced for the system to be within specifications.
8	DETERMINING_ORIENTATION	INS is determining the IMU axis aligned with gravity.
9	WAITING_INITIALPOS	The INS filter has determined the IMU orientation and is awaiting an initial position estimate to begin the alignment process.

a. This value is configured using the INSTHRESHOLDS command. See *INSTHRESHOLDS* on page 56.

5.2.16 INSATTS
Short INS Attitude

This log is the short header version of the *INSATT* log (page 135).

Message ID: 319

Log Type: Synch

Recommended Input:

```
log insattsa ontime 1
```

ASCII Example:

```
%INSATTSA,1541,487975.000;1541,487975.000549050,2.755452422,-4.127365126,323.289778434,
INS_SOLUTION_GOOD*ba08754f
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+12
5	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+20
6	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+28
7	Status	INS status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.17 INSATTX Inertial Attitude – Extended

This log includes the information from the INSATT log, as well as information about the attitude standard deviation. The position type and solution status fields indicate whether or not the corresponding data is valid.

> The INSATTX log is a large log and is not recommend for high rate logging.
> If you want to use high rate logging, log the INSATTS log at a high rate and the INSCOVS log ontime 1.

Message ID: **1457**

Log Type: **Synch**

Recommended Input:

```
log insattxa ontime 1
```

ASCII example:

```
#INSATTXA,COM1,0,81.0,FINESTEERING,1690,494542.000,00000040,5d25,43441;
INS_ALIGNMENT_COMPLETE,INS_PSRSP,1.137798832,-0.163068414,135.754208544,0.017797431,
0.017861038,3.168394804,4,0*f944b004
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	INSATTX Header	Log header		H	0
2	INS Status	Solution status See *Table 32, Inertial Solution Status* on page 136	Enum	4	H
3	Pos Type	Position type See *Table 33, Position or Velocity Type* on page 139	Enum	4	H+4
4	Roll	Roll in Local Level (degrees)	Double	8	H+8
5	Pitch	Pitch in Local Level (degrees)	Double	8	H+16
6	Azimuth	Azimuth in Local Level (degrees) This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+24
7	Roll σ	Roll standard deviation (degrees)	Float	4	H+32
8	Pitch σ	Pitch standard deviation (degrees)	Float	4	H+36
9	Azimuth σ	Azimuth standard deviation (degrees)	Float	4	H+40
10	Ext sol stat	Extended solution status See *Table 34, Extended Solution Status* on page 139	Hex	4	H+44
11	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+48
11	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+50
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 33: Position or Velocity Type

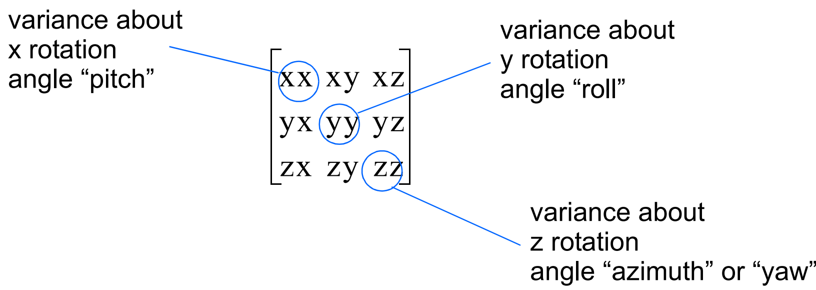
Binary	ASCII	Description
0	NONE	No Solution
1-51	Reserved	
52	INS_SBAS	INS SBAS solution
53	INS_PSRSP	INS pseudorange single point solution - no DGPS corrections
54	INS_PSRDIFF	INS pseudorange differential solution
55	INS_RTKFLOAT	INS RTK floating point ambiguities solution
56	INS_RTKFIXED	INS RTK fixed ambiguities solution
57	INS_OMNISTAR	INS OmniSTAR VBS position (L1 sub-meter)
58	INS_OMNISTAR_HP	INS OmniSTAR high precision solution
59	INS_OMNISTAR_XP	INS OmniSTAR extra precision solution
60-72	Reserved	
73	INS_PPP_Converging	INS NovAtel CORRECT Precise Point Positioning (PPP) solution converging
74	INS_PPP	INS NovAtel CORRECT Precise Point Positioning (PPP) solution

Table 34: Extended Solution Status

Bit	Mask	Description
0	0x00000001	A position update was applied in the last update epoch.
1	0x00000002	A phase update was applied in the last update epoch.
2	0x00000004	A ZUPT was applied in the last update epoch.
3	0x00000008	A wheel sensor update was applied in the last update epoch.
4	0x00000010	A heading (ALIGN) update was applied in the last update epoch.
5	0x00000020	Reserved
6	0x00000040	The INS solution has converged error estimates.
7	0x00000080 - 0x80000000	Reserved

5.2.18 INSCOV
INS Covariance Matrices

The position, attitude and velocity matrices in this log each contain 9 covariance values, with respect to the local level frame. For the attitude angles, they are given in the SPAN computation frame, as follows:



and are displayed within the log output as:

...,xx,xy,xz,yx,yy,yz,zx,zy,zz,...

These values are computed once per second and are available before and after alignment.

Message ID: 264

Log Type: Synch

Recommended Input:

```
log inscova ontime 1
```

ASCII Example:

```
#INSCOVA,COM1,0,65.0,FINESTEERING,1724,219604.009,00000040,0929,30019;1724,
219604.000000000,0.1285331446664655,0.0346617784498892,-0.1479079453018866,
0.0346617784498892,0.2902226803503227,0.2254840962138562,-0.1479079453018865,
0.2254840962138562,1.2153278719243952,0.0315677907853296,-0.0005084795762484,
0.0001477207864819,-0.0005084795762484,0.0251931017171569,0.0002612907385699,
0.0001477207864819,0.0002612907385699,0.0359258489923869,0.0030912934913378,
0.0008584993488541,-0.0048141355877257,0.0008584993488541,0.0074998390999675,
0.0071447656377662,-0.0048141355877257,0.0071447656377662,0.0300191236990451*7e3c6fb8
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Position Covariance	Position covariance matrix in local level frame (metres squared)	List of 9 Doubles	72	H+12
5	Attitude Covariance	Attitude covariance matrix in local level frame. (degrees squared – rotation around the given axis)	List of 9 Doubles	72	H+84
6	Velocity Covariance	Velocity covariance matrix in local level frame. (metres/second squared)	List of 9 Doubles	72	H+156
7	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+228
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

> The INS covariance values reported by the SPAN filter are an estimate of the Inertial filter solution quality. In lower accuracy GNSS position modes, such as SINGLE or WAAS (see *Table 25, Position or Velocity Type* on page 114), the position covariance values can appear to become optimistic compared with the absolute GNSS accuracy. This is due to the INS filter's ability to smooth short term noise in the GNSS solution, although the overall position error envelope still reflects the GNSS accuracy. Therefore, if the desired application requires absolute GNSS position accuracy, it is recommended to also monitor GNSS position messages such as BESTGNSSPOS (see *BESTGNSSPOS* on page 112).

5.2.19 INSCOVS
Short INS Covariance Log

This is the short header version of the *INSCOV* log (page 140). These values are computed once per second.

Message ID: **320**

Log Type: **Synch**

Recommended Input:

```
log inscovsa ontime 1
```

ASCII Example:

```
%INSCOVSA,1105,425385.020;1105,425385.000000000,0.0997319969301073,
-0.0240959791179416,-0.0133921499963209,-0.0240959791179416,0.1538605784734939,
0.0440068023663888,-0.0133921499963210,0.0440068023663887,0.4392033415009359,
0.0034190251365443,0.0000759398593357,-0.1362852812808768,0.0000759398593363,
0.0032413999569636,-0.0468473344270137,-0.1362852812808786,-0.0468473344270131,
117.5206493841025100,0.0004024901765302,-0.0000194916086028,0.0000036582459112,
-0.0000194916086028,0.0004518869575566,0.0000204616202028,0.0000036582459112,
0.0000204616202028,0.0005095575483948*1fc92787
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Position Covariance	Position covariance matrix in local level frame. (metres squared) xx,xy,xz,yx,yy,yz,zx,zy,zz	List of 9 Doubles	72	H+12
5	Attitude Covariance	Attitude covariance matrix of the SPAN frame to the local level frame. See page 140 for an example. (degrees squared – rotation around the given axis) xx,xy,xz,yx,yy,yz,zx,zy,zz	List of 9 Doubles	72	H+84
6	Velocity Covariance	Velocity covariance matrix in local level frame. (metres/second squared) xx,xy,xz,yx,yy,yz,zx,zy,zz	List of 9 Doubles	72	H+156
7	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+228
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

The INS covariance values reported by the SPAN filter are an estimate of the Inertial filter solution quality. In lower accuracy GNSS position modes, such as SINGLE or WAAS (see *Table 25, Position or Velocity Type* on page 114), the position covariance values can appear to become optimistic compared with the absolute GNSS accuracy. This is due to the INS filter's ability to smooth short term noise in the GNSS solution, although the overall position error envelope still reflects the GNSS accuracy. Therefore, if the desired application requires absolute GNSS position accuracy, it is recommended to also monitor GNSS position messages such as BESTGNSSPOS (see *BESTGNSSPOS* on page 112).

5.2.20 INSPOS
INS Position

This log contains the most recent position measurements in WGS84 coordinates and includes an INS status indicator. The log reports the position at the IMU center, unless the SETINSOFFSET command is issued, see *page 86*.

Message ID: 265

Log Type: Synch

Recommended Input:

```
log insposa ontime 1
```

ASCII Example:

```
#INSPOSA,USB2,0,18.0,FINESTEERING,1541,487977.000,00040000,17cd,37343;1541, 487977.
000549050,51.121315135,-114.042311349,1038.660737046,INS_SOLUTION_GOOD *2fffd557
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84)	Double	8	H+12
5	Longitude	Longitude (WGS84)	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84) [m]	Double	8	H+28
7	Status	INS status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.21 INSPOSS Short INS Position

This log is the short header version of the *INSPOS* log (page 143).

Message ID: 321

Log Type: Synch

Recommended Input:

```
log inspossa ontime 1
```

ASCII Example:

```
%INSPOSSA,1541,487916.000;1541,487916.000549050,51.115797277,-114.037811065,
1039.030700122,INS_SOLUTION_GOOD*5ca30894
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84)	Double	8	H+12
5	Longitude	Longitude (WGS84)	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84) [m]	Double	8	H+28
7	Status	INS status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.22 INSPOSX
Inertial Position – Extended

This log includes the information from the INSPOS log, as well as information about the position standard deviation. The position type and solution status fields indicate whether or not the corresponding data is valid.

> The INSPOSX log is a large log and is not recommend for high rate logging.
> If you want to use high rate logging, log the INSPOSS log at a high rate and the INSCOVS log ontime 1.

Message ID: **1459**

Log Type: **Synch**

Recommended Input:

```
log insposxa ontime 1
```

ASCII example:

```
#INSPOSXA,COM1,0,79.0,FINESTEERING,1690,493465.000,00000040,7211,43441;INS_SOLUTION_GOOD
,INS_PSRSP,51.11637750859,-114.03826206294,1049.1191,0.4883,0.4765,0.8853,3,0*dee048ab
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	INSPOSX Header	Log header		H	0
2	INS Status	Solution status See *Table 32, Inertial Solution Status* on page 136	Enum	4	H
3	Pos Type	Position type See *Table 33, Position or Velocity Type* on page 139	Enum	4	H+4
4	Lat	Latitude	Double	8	H+8
5	Long	Longitude	Double	8	H+16
6	Height	Orthometric height (m)	Double	8	H+24
7	Undulation	Undulation (m)	Float	4	H+32
8	Lat σ	Latitude standard deviation	Float	4	H+36
9	Long σ	Longitude standard deviation	Float	4	H+34
10	Height σ	Height standard deviation	Float	4	H+44
11	Ext sol stat	Extended solution status See *Table 34, Extended Solution Status* on page 139	Hex	4	H+48
11	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+52
12	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+54
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

> The INS standard deviation values reported by the SPAN filter are an estimate of the Inertial filter solution quality. In lower accuracy GNSS position modes, such as SINGLE or WAAS (see *Table 25, Position or Velocity Type* on page 114), the position standard deviation values can appear to become optimistic compared with the absolute GNSS accuracy. This is due to the INS filter's ability to smooth short term noise in the GNSS solution, although the overall position error envelope still reflects the GNSS accuracy. Therefore, if the desired application requires absolute GNSS position accuracy, it is recommended to also monitor GNSS position messages such as BESTGNSSPOS (see *BESTGNSSPOS* on page 112).

5.2.23 INSPVA
INS Position, Velocity and Attitude

This log allows INS position, velocity and attitude, with respect to the SPAN frame, to be collected in one log, instead of using three separate logs. Refer to *INSATT* on page 135 for an explanation of how the SPAN frame may differ from the IMU enclosure frame.

The attitude can be output in the vehicle frame. See *APPLYVEHICLEBODYROTATION* on page 36.

Message ID: 507

Log Type: Synch

Recommended Input:

```
log inspvaa ontime 1
```

ASCII Example:

```
#INSPVAA,COM1,0,31.0,FINESTEERING,1264,144088.000,00040000,5615,1541;
1264,144088.002284950,51.116827527,-114.037738908,401.191547167,354.846489850,
108.429407241,-10.837482850,1.116219952,-3.476059035,7.372686190,
INS_ALIGNMENT_COMPLETE*af719fd9
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84) [degrees]	Double	8	H+12
5	Longitude	Longitude (WGS84) [degrees]	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84) [m]	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction) [m/s]	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction) [m/s]	Double	8	H+44
9	Up Velocity	Velocity in an up direction [m/s]	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+76
13	Status	INS Status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.24 INSPVAS Short INS Position, Velocity and Attitude

This log is the short header version of the *INSPVA* log (page 146).

Message ID: **508**

Log Type: **Synch**

Recommended Input:

```
log inspvasa ontime 1
```

ASCII Example:

```
%INSPVASA,1264,144059.000;
1264,144059.002135700,51.116680071,-114.037929194,515.286704183,277.896368884,
84.915188605,-8.488207941,0.759619515,-2.892414901,6.179554750,INS_ALIGNMENT_COMPLETE
*855d6f76
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84) [degrees]	Double	8	H+12
5	Longitude	Longitude (WGS84) [degrees]	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84) [m]	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction) [m/s]	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction) [m/s]	Double	8	H+44
9	Up Velocity	Velocity in an up direction [m/s]	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from north This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+76
13	Status	INS Status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.25 INSPVAX
Inertial PVA – Extended

This log includes the information from the INSPVA log, as well as information about the position standard deviation. The position type and solution status fields indicate whether or not the corresponding data is valid.

> The INSPVAX log is a large log and is not recommend for high rate logging.
> If you want to use high rate logging, log the INSPVAS log at a high rate and the INSCOVS log ontime 1.

Message ID: 1465

Log Type: Synch

Recommended Input:

```
log inspvaxa ontime 1
```

ASCII example:

```
#INSPVAXA,COM1,0,73.5,FINESTEERING,1695,309428.000,00000040,4e77,43562;
INS_SOLUTION_GOOD,INS_PSRSP,51.11637873403,-114.03825114994,1063.6093,-16.9000,
-0.0845,-0.0464,-0.0127,0.138023492,0.069459386,90.000923268,0.9428,0.6688,1.4746,
0.0430,0.0518,0.0521,0.944295466,0.944567084,1.000131845,3,0*e877c178
```

Field	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	INSPVAX Header	Log header		H	0
2	INS Status	Solution status See *Table 32, Inertial Solution Status* on page 136	Enum	4	H
3	Pos Type	Position type See *Table 33, Position or Velocity Type* on page 139	Enum	4	H+4
4	Lat	Latitude (degrees)	Double	8	H+8
5	Long	Longitude (degrees)	Double	8	H+16
6	Height	Orthometric height (m)	Double	8	H+24
7	Undulation	Undulation (m)	Float	4	H+32
8	North Vel	North velocity (m/s)	Double	8	H+36
9	East Vel	East velocity (m/s)	Double	8	H+44
10	Up Vel	Up velocity (m/s)	Double	8	H+52
11	Roll	Roll in Local Level (degrees)	Double	8	H+60
12	Pitch	Pitch in Local Level (degrees)	Double	8	H+68
13	Azimuth	Azimuth in Local Level (degrees) This is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	Double	8	H+76
14	Lat σ	Latitude standard deviation (m)	Float	4	H+84
15	Long σ	Longitude standard deviation (m)	Float	4	H+88
16	Height σ	Height standard deviation (m)	Float	4	H+92
17	North Vel σ	North velocity standard deviation (m/s)	Float	4	H+96

Field	Field Type	Data Description	Format	Binary Bytes	Binary Offset
18	East Vel σ	East velocity standard deviation (m/s)	Float	4	H+100
19	Up Vel σ	Up velocity standard deviation (m/s)	Float	4	H+104
20	Roll σ	Roll standard deviation (degrees)	Float	4	H+108
21	Pitch σ	Pitch standard deviation (degrees)	Float	4	H+112
22	Azimuth σ	Azimuth standard deviation (degrees)	Float	4	H+116
23	Ext sol stat	Extended solution status See *Table 34, Extended Solution Status* on page 139	Hex	4	H+120
24	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+124
25	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+126
26	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

The INS standard deviation values reported by the SPAN filter are an estimate of the Inertial filter solution quality. In lower accuracy GNSS position modes, such as SINGLE or WAAS (see *Table 25, Position or Velocity Type* on page 114), the position standard deviation values can appear to become optimistic compared with the absolute GNSS accuracy. This is due to the INS filter's ability to smooth short term noise in the GNSS solution, although the overall position error envelope still reflects the GNSS accuracy. Therefore, if the desired application requires absolute GNSS position accuracy, it is recommended to also monitor GNSS position messages such as BESTGNSSPOS (see *BESTGNSSPOS* on page 112).

5.2.26 INSSPD INS Speed

This log contains the most recent speed measurements in the horizontal and vertical directions and includes an INS status indicator.

Message ID: 266

Log Type: Synch

Recommended Input:

```
log insspda ontime 1
```

ASCII Example:

```
#INSSPDA,USB2,0,20.0,FINESTEERING,1541,487969.000,00040000,7832,37343;1541,487969.000549
050,329.621116190,14.182070674,-0.126606551,INS_SOLUTION_GOOD *c274fff2
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Trk gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees The track over ground is determined by comparing the current position determined from the GNSS/INS solution with the previously determined position. Track over ground is best used when the vehicle is moving. When the vehicle is stationary, position error can make the direction of motion appear to change randomly.	Double	8	H+12
5	Horizontal Speed	Magnitude of horizontal speed in m/s where a positive value indicates forward movement and a negative value indicates reverse movement.	Double	8	H+20
6	Vertical Speed	Magnitude of vertical speed in m/s where a positive value indicates speed upward and a negative value indicates speed downward.	Double	8	H+28
7	Status	INS status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.27 INSSPDS Short INS Speed

This log is the short header version of the *INSSPD* log (page 150).

Message ID: 323

Log Type: Synch

Recommended Input:

```
log insspdsa ontime 1
```

ASCII Example:

```
%INSSPDSA,1541,487975.000;1541,487975.000549050,323.101450813,9.787233999,-0.038980077,
INS_SOLUTION_GOOD*105ba028
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Trk gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees. The track over ground is determined by comparing the current position determined from the GNSS/INS solution with the previously determined position. Track over ground is best used when the vehicle is moving. When the vehicle is stationary, position error can make the direction of motion appear to change randomly.	Double	8	H+12
5	Horizontal Speed	Magnitude of horizontal speed in m/s where a positive value indicates forward movement and a negative value indicates reverse movement.	Double	8	H+20
6	Vertical Speed	Magnitude of vertical speed in m/s where a positive value indicates speed upward and a negative value indicates speed downward.	Double	8	H+28
7	Status	INS status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.28 INSUPDATE
INS Update

This log contains the most recent INS update information. It provides information about what updates were performed in the INS filter at the last update epoch and a wheel sensor status indicator.

Message ID: 757

Log Type: Asynch

Recommended Input:

```
log insupdatea onchanged
```

ASCII Example:

```
#INSUPDATEA,SPECIAL,0,48.0,FINESTEERING,1701,156566.000,00004000,6f07,10883;SINGLE,0,12,
0,FALSE,WHEEL_SENSOR_INACTIVE,HEADING_UPDATE_ACTIVE*553bef65
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Solution Type	Type of GNSS solution used for the last update, see *Table 25, Position or Velocity Type* on page 114	Enum	4	H
3	Reserved		Integer	4	H+4
4	#Phase	Number of raw phase observations used in the last INS filter update	Integer	4	H+8
5	Reserved		Integer	4	H+12
6	Zupt Flag	A zero velocity update was performed during the last INS filter update: 0 = False 1 = True	Boolean	4	H+16
7	Wheel Status	Wheel status 0 = INACTIVE 1 = ACTIVE 2 = USED 3 = UNSYNCED 4 = BAD_MISC 5 = HIGH_ROTATION	Enum	4	H+20
8	Heading Update	Status of the heading update during the last INS filter update. See *Table 35, Heading Update Values* on page 153.	Enum	4	H+24
9	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+28
10	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 35: Heading Update Values

Binary	ASCII	Description
0	INACTIVE	A heading update was not available.
1	ACTIVE	Heading updates are running, but the epoch is not used as an update. When all other rejection criteria pass, a heading update will still only be applied once every 5 seconds (20 seconds when stationary).
2	USED	The update for that epoch was taken.

5.2.29 INSVEL
INS Velocity

This log contains the most recent North, East and Up velocity vector values, with respect to the local level frame and also includes an INS status indicator.

Message ID: 267

Log Type: Synch

Recommended Input:

```
log insvela ontime 1
```

ASCII Example:

```
#INSVELA,USB1,0,19.0,FINESTEERING,1543,236173.000,00000000,9c95,37343;1543,236173.002500
000,14.139471871,-0.070354464,-0.044204369,INS_SOLUTION_GOOD *3c37c0fc
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	North Velocity	Velocity North in m/s	Double	8	H+12
5	East Velocity	Velocity East in m/s	Double	8	H+20
6	Up Velocity	Velocity Up in m/s	Double	8	H+28
7	Status	INS status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.30 INSVELS Short INS Velocity

This log is the short header version of the *INSVEL* log (page 154).

Message ID: 324

Log Type: Synch

Recommended Input:

```
log insvelsa ontime 1
```

ASCII Example:

```
%INSVELSA,1921,152855.200;1921,152855.200000000,0.1077,-9.8326,-0.1504,INS_SOLUTION_GOOD
*efd71f65
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	North Velocity	Velocity North m/s	Double	8	H+12
5	East Velocity	Velocity East m/s	Double	8	H+20
6	Up Velocity	Velocity Up m/s	Double	8	H+28
7	Status	INS status, see *Table 32, Inertial Solution Status* on page 136	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.31 INSVELX
Inertial Velocity – Extended

This log includes the information from the INSVEL log, as well as information about the velocity standard deviation. The position type and solution status fields indicate whether or not the corresponding data is valid.

> The INSVELX log is a large log and is not recommend for high rate logging.
> If you want to use high rate logging, log the INSVELS log at a high rate and the INSCOVS log ontime 1.

Message ID: 1458

Log Type: Synch

Recommended Input:

```
log insvelxa ontime 1
```

ASCII example:

```
#INSVELXA,COM1,0,80.0,FINESTEERING,1690,494394.000,00000040,1f8e,43441;
INS_ALIGNMENT_COMPLETE,INS_PSRSP,0.0086,0.0015,0.0215,0.0549,0.0330,0.0339,3,0*ec33e372
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	INSVELX Header	Log header		H	0
2	INS Status	Solution status See *Table 32, Inertial Solution Status* on page 136	Enum	4	H
3	Pos Type	Position type See *Table 33, Position or Velocity Type* on page 139	Enum	4	H+4
4	North Vel	North velocity (m/s)	Double	8	H+8
5	East Vel	East velocity (m/s)	Double	8	H+16
6	Up Vel	Up velocity (m/s)	Double	8	H+24
7	North Vel σ	North velocity standard deviation (m/s)	Float	4	H+32
8	East Vel σ	East velocity standard deviation (m/s)	Float	4	H+36
9	Up Vel σ	Up velocity standard deviation (m/s)	Float	4	H+40
10	Ext sol stat	Extended solution status See *Table 34, Extended Solution Status* on page 139	Hex	4	H+44
11	Time Since Update	Elapsed time since the last ZUPT or position update (seconds)	Ushort	2	H+48
11	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+50
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.32 MARK1COUNT Count for Mark1 Input

This log is only available for SPAN systems with an OEM638 or ProPak6 receiver.

This log contains the tick count for the EVENT1 input.

When the input mode is set to COUNT using the EVENTINCONTROL command, see *page 43*, the MARKxCOUNT logs become available.

1. Use the ONNEW trigger with this, the MARK*x*TIME, or the MARK*x*PVA logs.
2. Only the MARK*x*COUNT, MARK*x*PVA logs, the MARK*x*TIME logs, and 'polled' log types are generated 'on the fly' at the exact time of the mark. Synchronous and asynchronous logs output the most recently available data.

Message ID: 1093

Log Type: Asynch

Recommended Input:

```
log mark1counta onnew
```

ASCII Example:

```
#MARK1COUNTA,COM1,0,98.5,FINESTEERING,1520,515353.000,00000000,0000,137;
1000000,1*1786750b
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	MARK1COUNT header	Log header		H	0
2	Period	Delta time	Ulong	4	H
3	Count	Tick count	Ushort	2	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+6
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.33 MARK1PVA Position, Velocity and Attitude at Mark1

This log outputs position, velocity and attitude information, with respect to the SPAN frame, when an event is received on the Mark1 input. If the `SETMARK1OFFSET` command has been entered, the MARK1PVA log will contain the solution translated, and then rotated, by the values provided in the command. See the `SETMARK1OFFSET` command, valid at the time, on page 89.

Message ID: 1067

Log Type: Synch

Recommended Input:

```
log mark1pva onnew
```

Abbreviated ASCII Example:

```
#MARK1PVAA,COM1,0,74.5,FINESTEERING,1732,247231.455,00040020,5790,12002;
1732,247231.454623850,51.11693182283,-114.03885213810,1047.4525,0.0004,0.0004,
-0.0006,0.847121689,1.124640813,278.577037489,INS_SOLUTION_GOOD*5a6b060e
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week at Mark1 request	Ulong	4	H
3	Seconds	Seconds from week at Mark1	Double	8	H+4
4	Latitude	Latitude (WGS84) at Mark1	Double	8	H+12
5	Longitude	Longitude (WGS84) at Mark1	Double	8	H+20
6	Height	Height (WGS84) at Mark1	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction) at Mark1	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction) at Mark1	Double	8	H+44
9	Up Velocity	Velocity in an up direction at Mark1	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees at Mark1	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees at Mark1	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from North at Mark1	Double	8	H+76
13	Status	INS Status, see *Table 32, Inertial Solution Status* on page 136 at Mark1	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.34 MARK2COUNT
Count for Mark2 Input

> This log is only available for SPAN systems with an OEM638 or ProPak6 receiver.

This log contains the tick count for the EVENT2 input.

When the input mode is set to COUNT using the `EVENTINCONTROL` command, see *page 43*, the MARKxCOUNT logs become available.

> 1. Use the ONNEW trigger with this, the MARK*x*TIME, or the MARK*x*PVA logs.
> 2. Only the MARK*x*COUNT, MARK*x*PVA logs, the MARK*x*TIME logs, and 'polled' log types are generated 'on the fly' at the exact time of the mark. Synchronous and asynchronous logs output the most recently available data.

Message ID: 1094

Log Type: Asynch

Recommended Input:

```
log mark2counta onnew
```

ASCII Example:

```
#MARK2COUNTA,COM1,0,98.5,FINESTEERING,1520,515353.000,00000000,0000,137;
1000000,1*1786750b
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	MARK2COUNT header	Log header		H	0
2	Period	Delta time	Ulong	4	H
3	Count	Tick count	Ushort	2	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+6
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.35 MARK2PVA Position, Velocity and Attitude at Mark2

This log outputs position, velocity and attitude information, with respect to the SPAN frame, when an event was received on the Mark2 input. If the `SETMARK2OFFSET` command has been entered, the MARK2PVA log will contain the solution translated, and then rotated, by the values provided in the command. See the `SETMARK2OFFSET` command, valid at the time, on page 90.

> The SPAN-CPT does not have a Mark2 input, so this log is not available for the SPAN-CPT.

Message ID: 1068

Log Type: Synch

Recommended Input:

```
log mark2pva onnew
```

Abbreviated ASCII Example:

```
#MARK2PVAA,COM1,0,74.5,FINESTEERING,1732,247232.271,00040020,2425,12002;
1732,247232.271459820,51.11693179023,-114.03885206704,1047.4529,0.0004,-0.0011,
-0.0007,0.837101074,1.134127754,278.346498557,INS_SOLUTION_GOOD*08209ec0
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week at Mark2 request	Ulong	4	H
3	Seconds	Seconds from week at Mark2	Double	8	H+4
4	Latitude	Latitude (WGS84) at Mark2	Double	8	H+12
5	Longitude	Longitude (WGS84) at Mark2	Double	8	H+20
6	Height	Height (WGS84) at Mark2	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction) at Mark2	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction) at Mark2	Double	8	H+44
9	Up Velocity	Velocity in an up direction at Mark2	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees at Mark2	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees at Mark2	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from north at Mark2	Double	8	H+76
13	Status	INS Status, see *Table 32, Inertial Solution Status* on page 136 at Mark2	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.36 MARK3COUNT Count for Mark3 Input

This log is only available for SPAN systems with an OEM638 or ProPak6 receiver.

When the input mode is set to COUNT using the `EVENTINCONTROL` command, see *page 43*, the MARKxCOUNT logs become available.

1. Use the ONNEW trigger with this, the MARK*x*TIME, or the MARK*x*PVA logs.
2. Only the MARK*x*COUNT, MARK*x*PVA logs, the MARK*x*TIME logs, and 'polled' log types are generated 'on the fly' at the exact time of the mark. Synchronous and asynchronous logs output the most recently available data.

Message ID: 1095

Log Type: Asynch

Recommended Input:

```
log mark3counta onnew
```

ASCII Example:

```
#MARK3COUNTA,COM1,0,98.5,FINESTEERING,1520,515353.000,00000000,0000,137;
1000000,1*1786750b
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	MARK3COUNT header	Log header		H	0
2	Period	Delta time	Ulong	4	H
3	Count	Tick count	Ushort	2	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+6
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.37 MARK3PVA Position, Velocity and Attitude at Mark3

This log is only available for SPAN systems with an OEM638 or ProPak6 receiver.

This log outputs position, velocity and attitude information, with respect to the SPAN frame, when an event was received on the Mark3 input. If the SETMARK3OFFSET command has been entered, the MARK3PVA log will contain the solution translated, and then rotated, by the values provided in the command. See the SETMARK3OFFSET command, valid at the time, on page 91.

Message ID: 1118

Log Type: Synch

Recommended Input:

```
log mark3pva onnew
```

Abbreviated ASCII Example:

```
#MARK3PVAA,COM1,0,74.5,FINESTEERING,1732,247232.271,00040020,2425,12002;
1732,247232.271459820,51.11693179023,-114.03885206704,1047.4529,0.0004,-0.0011,
-0.0007,0.837101074,1.134127754,278.346498557,INS_SOLUTION_GOOD*08209ec0
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week at Mark3 request	Ulong	4	H
3	Seconds	Seconds from week at Mark3	Double	8	H+4
4	Latitude	Latitude (WGS84) at Mark3	Double	8	H+12
5	Longitude	Longitude (WGS84) at Mark3	Double	8	H+20
6	Height	Height (WGS84) at Mark3	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction) at Mark3	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction) at Mark3	Double	8	H+44
9	Up Velocity	Velocity in an up direction at Mark3	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees at Mark3	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees at Mark3	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from north at Mark3	Double	8	H+76
13	Status	INS Status, see *Table 32, Inertial Solution Status* on page 136 at Mark3	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.38 MARK4COUNT
Count for Mark4 Input

This log is only available for SPAN systems with an OEM638 or ProPak6 receiver.

When the input mode is set to COUNT using the EVENTINCONTROL command, see *page 43*, the MARKxCOUNT logs become available.

1. Use the ONNEW trigger with this, the MARK*x*TIME, or the MARK*x*PVA logs.
2. Only the MARK*x*COUNT, MARK*x*PVA logs, the MARK*x*TIME logs, and 'polled' log types are generated 'on the fly' at the exact time of the mark. Synchronous and asynchronous logs output the most recently available data.

Message ID: 1096

Log Type: Asynch

Recommended Input:

```
log mark4counta onnew
```

ASCII Example:

```
#MARK4COUNTA,COM1,0,98.5,FINESTEERING,1520,515353.000,00000000,0000,137;
1000000,1*1786750b
```

Field	Field type	Description	Format	Binary Bytes	Binary Offset
1	MARK4COUNT header	Log header		H	0
2	Period	Delta time	Ulong	4	H
3	Count	Tick count	Ushort	2	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+6
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.39 MARK4PVA
Position, Velocity and Attitude at Mark4

> This log is only available for SPAN systems with an OEM638 or ProPak6 receiver.

This log outputs position, velocity and attitude information, with respect to the SPAN frame, when an event was received on the Mark4 input. If the `SETMARK4OFFSET` command has been entered, the MARK4PVA log will contain the solution translated, and then rotated, by the values provided in the command. See the `SETMARK4OFFSET` command, valid at the time, on page 92.

Message ID: 1119

Log Type: Synch

Recommended Input:

```
log mark4pva onnew
```

Abbreviated ASCII Example:

```
#MARK4PVAA,COM1,0,74.5,FINESTEERING,1732,247232.271,00040020,2425,12002;
1732,247232.271459820,51.11693179023,-114.03885206704,1047.4529,0.0004,-0.0011,
-0.0007,0.837101074,1.134127754,278.346498557,INS_SOLUTION_GOOD*08209ec0
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week at Mark4 request	Ulong	4	H
3	Seconds	Seconds from week at Mark4	Double	8	H+4
4	Latitude	Latitude (WGS84) at Mark4	Double	8	H+12
5	Longitude	Longitude (WGS84) at Mark4	Double	8	H+20
6	Height	Height (WGS84) at Mark4	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction) at Mark4	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction) at Mark4	Double	8	H+44
9	Up Velocity	Velocity in an up direction at Mark4	Double	8	H+52
10	Roll	Right-handed rotation from local level around y-axis in degrees at Mark4	Double	8	H+60
11	Pitch	Right-handed rotation from local level around x-axis in degrees at Mark4	Double	8	H+68
12	Azimuth	Left-handed rotation around z-axis in degrees clockwise from north at Mark4	Double	8	H+76
13	Status	INS Status, see *Table 32, Inertial Solution Status* on page 136 at Mark4	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.40 PASHR
NMEA, Inertial Attitude Data

The PASHR log uses a UTC time, calculated with default parameters, to output NMEA messages without waiting for a valid almanac. The UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters and sets the UTC time to VALID. For more information about NMEA, refer to the *OEM6 Family Firmware Reference Manual* (OM-20000129). The PASHR log contains only INS derived attitude information and is only filled when an inertial solution is available.

Message ID: 1177

Log Type: Synch

Recommended Input:

```
log pashr ontime 1
```

Example:

```
$PASHR,,,,,,,,,,0*68 (empty)
$PASHR,195124.00,305.30,T,+0.05,-0.13,,0.180,0.185,4.986,1*2B
```

Field	Structure	Description	Symbol	Example
1	$PASHR	Log Header	---	$PASHR
2	Time	UTC Time	hhmmss.ss	195124.00
3	Heading	Heading value in decimal degrees The heading is the inertial azimuth calculated from the IMU gyros and the SPAN filters.	HHH.HH	305.30
4	True Heading	T displayed if heading is relative to true north.	T	T
5	Roll	Roll in decimal degrees. The ± sign will always be displayed.	RRR.RR	+0.05
6	Pitch	Pitch in decimal degrees. The ± sign will always be displayed.	PPP.PP	-0.13
7	Reserved	------	----	----
8	Roll Accuracy	Roll standard deviation in decimal degrees.	rr.rrr	0.180
9	Pitch Accuracy	Pitch standard deviation in decimal degrees.	pp.ppp	0.185
10	Heading Accuracy	Heading standard deviation in decimal degrees.	hh.hhh	4.986
11	GPS Update Quality Flag	0 = No position 1 = All non-RTK fixed integer positions 2 = RTK fixed integer position	1	1
12	Checksum	Checksum	*XX	*2B
13	[CR][LF]	Sentence terminator		[CR][LF]

5.2.41 RAWIMU
Raw IMU Data

This log contains an IMU status indicator and the measurements from the accelerometers and gyros with respect to the IMU enclosure frame. If logging this data, consider the RAWIMUS log to reduce the amount of data, see page 179.

Message ID: 268

Log Type: Asynch

Recommended Input:

```
log rawimua onnew
```

ASCII Example:

```
#RAWIMUA,COM1,0,68.5,FINESTEERING,1724,219418.009,004c0040,6125,30019;1724,
219418.008755000,00000077,64732,56,298,8,28,-3*7378486f
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	IMU Status	The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs. For the raw IMU status, see one of the following tables: • *Table 36, iIMU-FSAS IMU Status* on page 168 • *Table 37, Litef LCI-1 IMU Status* on page 169 • *Table 39, HG1700 IMU Status* on page 170 • *Table 40, LN200 IMU Status* on page 171 • *Table 41, ISA-100 and ISA-100C IMU Status* on page 172 • *Table 42, IMU-CPT and SPAN-CPT IMU Status* on page 173 • *Table 43, IMU-KVH1750 and IMU-KVH1725 IMU Status* on page 174 • *Table 44, HG1900 and HG1930 IMU Status* on page 175 • *Table 45, ADIS16488, IMU-IGM-A1 and SPAN-IGM-A1 IMU Status* on page 176 • *Table 46, STIM300, IMU-IGM-S1, SPAN-IGM-S1 IMU Status* on page 177 • *Table 47, µIMU IMU Status* on page 178 Also refer to Interface Control Documentation as provided by Honeywell or Northrop Grumman.	Hex Ulong	4	H+12
5	Z Accel Output	Change in velocity count along z axis [a]	Long	4	H+16
6	- (Y Accel Output)	- (Change in velocity count along y axis) [a, b]	Long	4	H+20

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	X Accel Output	Change in velocity count along x axis [a]	Long	4	H+24
8	Z Gyro Output	Change in angle count around z axis [c]. Right-handed	Long	4	H+28
9	- (Y Gyro Output)	- (Change in angle count around y axis) [b, c]. Right-handed	Long	4	H+32
10	X Gyro Output	Change in angle count around x axis [c]. Right-handed	Long	4	H+36
11	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. The change in velocity (acceleration) scale factor for each IMU type can be found in *Table 48, Raw IMU Scale Factors* on page 180. Multiply the scale factor in *Table 48, Raw IMU Scale Factors* by the count in this field for the velocity increments.

b. A negative value implies the output is along the positive y-axis marked on the IMU.
A positive value implies the change is in the direction opposite to that of the y-axis marked on the IMU.

c. The change in angle (gyro) scale factor can be found in *Table 48, Raw IMU Scale Factors* on page 180. Multiply the appropriate scale factor in *Table 48, Raw IMU Scale Factors* by the count in this field for the angle increments in radians. To obtain acceleration in m/s^2, multiply the velocity increments by the output rate of the IMU (e.g., 100 Hz for HG1700, HG1900 and HG1930; 200 Hz ISA-100, ISA-100C, iMAR-FSAS, LN200, LCI-1, µIMU, KVH1750 and ADIS16488; 125 Hz for STIM300).

Table 36: iIMU-FSAS IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Reserved	
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	Gyro warm-up	0 = Passed, 1 = Failed
	5	0x00000020	Gyro self-test active	0 = Passed, 1 = Failed
	6	0x00000040	Gyro status bit set	0 = Passed, 1 = Failed
	7	0x00000080	Gyro time-out command interface	0 = Passed, 1 = Failed
N2	8	0x00000100	Power-up built-in test (PBIT)	0 = Passed, 1 = Failed
	9	0x00000200	Reserved	
	10	0x00000400	Interrupt	0 = Passed, 1 = Failed
	11	0x00000800	Reserved	
N3	12	0x00001000	Warm-up	0 = Passed, 1 = Failed
	13	0x00002000	Reserved	
	14	0x00004000		
	15	0x00008000	Initiated built-in test (IBIT)	0 = Passed, 1 = Failed
N4	16	0x00010000	Reserved	
	17	0x00020000		
	18	0x00040000	Accelerometer	0 = Passed, 1 = Failed
	19	0x00080000	Accelerometer time-out	0 = Passed, 1 = Failed
N5	20	0x00100000	Reserved	
	21	0x00200000	Gyro initiated BIT	0 = Passed, 1 = Failed
	22	0x00400000	Gyro self-test	0 = Passed, 1 = Failed
	23	0x00800000	Gyro time-out	0 = Passed, 1 = Failed
N6	24	0x01000000	Analog-to-Digital (AD)	0 = Passed, 1 = Failed
	25	0x02000000	Test mode	0 = Passed, 1 = Failed
	26	0x04000000	Software	0 = Passed, 1 = Failed
	27	0x08000000	RAM/ROM	0 = Passed, 1 = Failed
N7	28	0x10000000	Reserved	
	29	0x20000000	Operational	0 = Passed, 1 = Failed
	30	0x40000000	Interface	0 = Passed, 1 = Failed
	31	0x80000000	Interface time-out	0 = Passed, 1 = Failed

Table 37: Litef LCI-1 IMU Status

<table>
<tr><th>Nibble #</th><th>Bit #</th><th>Mask</th><th>Description</th><th colspan="2">Range Value</th></tr>
<tr><td rowspan="4">N0</td><td>0</td><td>0x00000001</td><td>IBIT Error Flag</td><td colspan="2">0 = Normal, 1 = IBIT Error</td></tr>
<tr><td>1</td><td>0x00000002</td><td>CBIT Error Flag</td><td colspan="2">0 = Normal, 1 = CBIT Error</td></tr>
<tr><td>2</td><td>0x00000004</td><td>Calibration Status Flag</td><td colspan="2">0 = IMU Uncalibrated, 1 = IMU Calibrated</td></tr>
<tr><td>3</td><td>0x00000008</td><td colspan="3">Not used</td></tr>
<tr><td rowspan="4">N1</td><td>4</td><td>0x00000010</td><td>Mode Read Flag</td><td>Mode in Trans.: 0,
Mode Trans Ready = 1</td><td rowspan="4">Refer to Table 38, Mode Indication on page 169</td></tr>
<tr><td>5</td><td>0x00000020</td><td>IMU Mode Indication 1</td><td>0 = Not Set, 1 = Set</td></tr>
<tr><td>6</td><td>0x00000040</td><td>IMU Mode Indication 2</td><td>0 = Not Set, 1 = Set</td></tr>
<tr><td>7</td><td>0x00000080</td><td>IMU Mode Indication 3</td><td>0 = Not Set, 1 = Set</td></tr>
<tr><td rowspan="4">N2</td><td>8</td><td>0x00000100</td><td>Master NoGo</td><td colspan="2">0 = Normal, 1 = NoGo</td></tr>
<tr><td>9</td><td>0x00000200</td><td>IMU NoGo</td><td colspan="2">0 = Normal, 1 = NoGo</td></tr>
<tr><td>10</td><td>0x00000400</td><td>Accelerometer Z NoGo</td><td colspan="2">0 = Normal, 1 = NoGo</td></tr>
<tr><td>11</td><td>0x00000800</td><td>Accelerometer Y NoGo</td><td colspan="2">0 = Normal, 1 = NoGo</td></tr>
<tr><td rowspan="4">N3</td><td>12</td><td>0x00001000</td><td>Accelerometer X NoGo</td><td colspan="2">0 = Normal, 1 = NoGo</td></tr>
<tr><td>13</td><td>0x00002000</td><td>Gyroscope Z NoGo</td><td colspan="2">0 = Normal, 1 = NoGo</td></tr>
<tr><td>14</td><td>0x00004000</td><td>Gyroscope Y NoGo</td><td colspan="2">0 = Normal, 1 = NoGo</td></tr>
<tr><td>15</td><td>0x00008000</td><td>Gyroscope X NoGo</td><td colspan="2">0 = Normal, 1 = NoGo</td></tr>
<tr><td rowspan="4">N4</td><td>16</td><td>0x00010000</td><td>Master Warning</td><td colspan="2">0 = Normal, 1 = Warning</td></tr>
<tr><td>17</td><td>0x00020000</td><td>IMU Warning</td><td colspan="2">0 = Normal, 1 = Warning</td></tr>
<tr><td>18</td><td>0x00040000</td><td>Accelerometer Z Warning</td><td colspan="2">0 = Normal, 1 = Warning</td></tr>
<tr><td>19</td><td>0x00080000</td><td>Accelerometer Y Warning</td><td colspan="2">0 = Normal, 1 = Warning</td></tr>
<tr><td rowspan="4">N5</td><td>20</td><td>0x00100000</td><td>Accelerometer X Warning</td><td colspan="2">0 = Normal, 1 = Warning</td></tr>
<tr><td>21</td><td>0x00200000</td><td>Gyroscope Z Warning</td><td colspan="2">0 = Normal, 1 = Warning</td></tr>
<tr><td>22</td><td>0x00400000</td><td>Gyroscope Y Warning</td><td colspan="2">0 = Normal, 1 = Warning</td></tr>
<tr><td>23</td><td>0x00800000</td><td>Gyroscope X Warning</td><td colspan="2">0 = Normal, 1 = Warning</td></tr>
<tr><td rowspan="4">N6</td><td>24</td><td>0x01000000</td><td colspan="3" rowspan="8">Not Used</td></tr>
<tr><td>25</td><td>0x02000000</td></tr>
<tr><td>26</td><td>0x04000000</td></tr>
<tr><td>27</td><td>0x08000000</td></tr>
<tr><td rowspan="4">N7</td><td>28</td><td>0x10000000</td></tr>
<tr><td>29</td><td>0x20000000</td></tr>
<tr><td>30</td><td>0x40000000</td></tr>
<tr><td>31</td><td>0x80000000</td></tr>
</table>

Table 38: Mode Indication

MDI3	MDI2	MDI1	MRF	Current IMU Mode
0	0	0	0	Power On BIT (PBIT)
0	0	0	1	Standby Mode
0	1	1	0	Initiated BIT (IBIT)
0	1	1	1	IBIT Ready
1	1	0	1	Operational Mode

Table 39: HG1700 IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Reserved	
	1	0x00000002	Reserved	
	2	0x00000004	Reserved	
	3	0x00000008	Reserved	
N1	4	0x00000010	IMU Status	0 = Passed, 1 = Failed
	5	0x00000020	IMU Status	0 = Passed, 1 = Failed
	6	0x00000040	IMU Status	0 = Passed, 1 = Failed
	7	0x00000080	IMU Status	0 = Passed, 1 = Failed
N2	8	0x00000100	Reserved	
	9	0x00000200	Reserved	
	10	0x00000400	Reserved	
	11	0x00000800	Reserved	
N3	12	0x00001000	Reserved	
	13	0x00002000	Reserved	
	14	0x00004000	Reserved	
	15	0x00008000	Reserved	
N4	16	0x00010000	Reserved	
	17	0x00020000	Reserved	
	18	0x00040000	Reserved	
	19	0x00080000	Reserved	
N5	20	0x00100000	Reserved	
	21	0x00200000	Reserved	
	22	0x00400000	Reserved	
	23	0x00800000	Reserved	
N6	24	0x01000000	Reserved	
	25	0x02000000	Reserved	
	26	0x04000000	Reserved	
	27	0x08000000	IMU Status	0 = Passed, 1= Failed
N7	28	0x10000000	IMU Status	0 = Passed, 1 = Failed
	29	0x20000000	IMU Status	0 = Passed, 1 = Failed
	30	0x40000000	IMU Status	0 = Passed, 1 = Failed
	31	0x80000000	IMU Status	0 = Passed, 1 = Failed

Table 40: LN200 IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	IMU Status	0 = Passed, 1 = Failed
	1	0x00000002	IMU Status	0 = Passed, 1 = Failed
	2	0x00000004	IMU Status	0 = Passed, 1 = Failed
	3	0x00000008	IMU Status	0 = Passed, 1 = Failed
N1	4	0x00000010	IMU Status	0 = Passed, 1 = Failed
	5	0x00000020	IMU Status	0 = Passed, 1 = Failed
	6	0x00000040	IMU Status	0 = Passed, 1 = Failed
	7	0x00000080	IMU Status	0 = Passed, 1 = Failed
N2	8	0x00000100	IMU Status	0 = Passed, 1 = Failed
	9	0x00000200	IMU Status	0 = Passed, 1 = Failed
	10	0x00000400	IMU Status	0 = Passed, 1 = Failed
	11	0x00000800	IMU Status	0 = Passed, 1 = Failed
N3	12	0x00001000	IMU Status	0 = Passed, 1 = Failed
	13	0x00002000	IMU Status	0 = Passed, 1 = Failed
	14	0x00004000	IMU Status	0 = Passed, 1 = Failed
	15	0x00008000	Reserved	
N4	16	0x00010000	Reserved	
	17	0x00020000	Reserved	
	18	0x00040000	Reserved	
	19	0x00080000	Reserved	
N5	20	0x00100000	Reserved	
	21	0x00200000	Reserved	
	22	0x00400000	Reserved	
	23	0x00800000	Reserved	
N6	24	0x01000000	IMU Status	0 = Passed, 1 = Failed
	25	0x02000000	IMU Status	0 = Passed, 1 = Failed
	26	0x04000000	IMU Status	0 = Passed, 1 = Failed
	27	0x08000000	IMU Status	0 = Passed, 1 = Failed
N7	28	0x10000000	IMU Status	0 = Passed, 1 = Failed
	29	0x20000000	Reserved	
	30	0x40000000	IMU Status	0 = Passed, 1 = Failed
	31	0x80000000	Reserved	

Table 41: ISA-100 and ISA-100C IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Maintenance Indication	0 = Normal, 1 = System Maintenance Indicator
	1	0x00000002	Accelerometers Invalid	0 = Normal, 1 = Invalid
	2	0x00000004	Accelerometer X Warning	0 = Normal, 1 = Warning
	3	0x00000008	Accelerometer Y Warning	0 = Normal, 1 = Warning
N1	4	0x00000010	Accelerometer Z Warning	0 = Normal, 1 = Warning
	5	0x00000020	Accelerometer X NOGO	0 = Normal, 1 = NOGO
	6	0x00000040	Accelerometer Y NOGO	0 = Normal, 1 = NOGO
	7	0x00000080	Accelerometer Z NOGO	0 = Normal, 1 = NOGO
N2	8	0x00000100	Reset Occurred	0 = Normal, 1 = First Message after ISA-100C Reset
	9	0x00000200	Gyroscopes Invalid	0 = Normal, 1 = Invalid
	10	0x00000400	Gyroscope X Warning	0 = Normal, 1 = Warning
	11	0x00000800	Gyroscope Y Warning	0 = Normal, 1 = Warning
N3	12	0x00001000	Gyroscope Z Warning	0 = Normal, 1 = Warning
	13	0x00002000	Gyroscope X NOGO	0 = Normal, 1 = NOGO
	14	0x00004000	Gyroscope Y NOGO	0 = Normal, 1 = NOGO
	15	0x00008000	Gyroscope Z NOGO	0 = Normal, 1 = NOGO
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte value (SHORT) 1 LSB = $3.90625e^{-3}$ Celsius Temperature Range +/- 128 Celsius	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 42: IMU-CPT and SPAN-CPT IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Gyro X Status	1 = Valid, 0 = Invalid
	1	0x00000002	Gyro Y Status	1 = Valid, 0 = Invalid
	2	0x00000004	Gyro Z Status	1 = Valid, 0 = Invalid
	3	0x00000008	Unused	Set to 0
N1	4	0x00000010	Accelerometer X Status	1 = Valid, 0 = Invalid
	5	0x00000020	Accelerometer Y Status	1 = Valid, 0 = Invalid
	6	0x00000040	Accelerometer Z Status	1 = Valid, 0 = Invalid
	7	0x00000080	Unused	Set to 0
N2	8	0x00000100	IMU Data Sequence Counter read in a Ushort.	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000		
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000	Unused	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 43: IMU-KVH1750 and IMU-KVH1725 IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Gyro X Status	1 = Valid, 0 = Invalid
	1	0x00000002	Gyro Y Status	1 = Valid, 0 = Invalid
	2	0x00000004	Gyro Z Status	1 = Valid, 0 = Invalid
	3	0x00000008	Unused	Set to 0
N1	4	0x00000010	Accelerometer X Status	1 = Valid, 0 = Invalid
	5	0x00000020	Accelerometer Y Status	1 = Valid, 0 = Invalid
	6	0x00000040	Accelerometer Z Status	1 = Valid, 0 = Invalid
	7	0x00000080	Unused	Set to 0
N2	8	0x00000100	IMU Data Sequence Counter read in a Ushort.	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000		
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte value (SHORT) Rounded to the nearest degree Example: <RAWIMU COM1 0 75.0 FINESTEERING 1813 514207.000 00000020 fa9a 45836 1813 514207.000000000 00260077 32164 -47 -305 1 -10 0 IMU status = 00260077 Temperatures bytes = 0026 Decimal value = 38 degrees C	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 44: HG1900 and HG1930 IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Reserved	
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	IMU Status	0 = Passed, 1 = Failed
	5	0x00000020	IMU Status	0 = Passed, 1 = Failed
	6	0x00000040	IMU Status	0 = Passed, 1 = Failed
	7	0x00000080	IMU Status	0 = Passed, 1 = Failed
N2	8	0x00000100	Reserved	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000	Reserved	
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000	Reserved	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000	Reserved	
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000	IMU Status	0 = Passed, 1 = Failed
	25	0x02000000	Reserved	
	26	0x04000000	IMU Status	0 = Passed, 1 = Failed
	27	0x08000000	IMU Status	0 = Passed, 1 = Failed
N7	28	0x10000000	IMU Status	0 = Passed, 1 = Failed
	29	0x20000000	IMU Status	0 = Passed, 1 = Failed
	30	0x40000000	IMU Status	0 = Passed, 1 = Failed
	31	0x80000000	Reserved	

Table 45: ADIS16488, IMU-IGM-A1 and SPAN-IGM-A1 IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Alarm Status Flag	
	1	0x00000002	Reserved	
	2	0x00000004		
	3	0x00000008	SPI Communication Error	0 = Passed, 1 = Failed
N1	4	0x00000010	Sensor Over-Range	0 = Passed, 1 = One of more sensors over-ranged
	5	0x00000020	Initial Self Test Failure	0 = Passed, 1= Failed
	6	0x00000040	Flash Memory Failure	0 = Passed, 1 = Failed
	7	0x00000080	Processing Overrun	0 = Passed, 1 = Failed
N2	8	0x00000100	Self Test Failure – X-axis gyro	0 = Passed, 1 = Failed
	9	0x00000200	Self Test Failure – Y-axis gyro	0 = Passed, 1 = Failed
	10	0x00000400	Self Test Failure – Z-axis gyro	0 = Passed, 1 = Failed
	11	0x00000800	Self Test Failure – X-axis accelerometer	0 = Passed, 1 = Failed
N3	12	0x00001000	Self Test Failure – Y-axis accelerometer	0 = Passed, 1 = Failed
	13	0x00002000	Self Test Failure – Z-axis	0 = Passed, 1 = Failed
	14	0x00004000	Reserved	
	15	0x00008000		
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte value (SHORT) 25°C = 0x0000 1 LSB = 0.00565°C	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 46: STIM300, IMU-IGM-S1, SPAN-IGM-S1 IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Gyro status	0 = OK, 1 = X-channel
	1	0x00000002		0 = OK, 1 = Y-channel
	2	0x00000004		0 = OK, 1 = Z-channel
	3	0x00000008		0 = OK, 1 = Error in measurement channel (Bits 0-2 flag the error channels)
N1	4	0x00000010		0 = OK, 1 = Overload (Bits 0-2 flag the error channels)
	5	0x00000020		0 = OK, 1 = Outside operating conditions
	6	0x00000040		0 = OK, 1 = Startup
	7	0x00000080		0 = OK, 1 = System integrity error
N2	8	0x00000100	Accelerometer Status	0 = OK, 1 = X-channel
	9	0x00000200		0 = OK, 1 = Y-channel
	10	0x00000400		0 = OK, 1 = Z-channel
	11	0x00000800		0 = OK, 1 = Error in measurement channel (Bits 0-2 flag the error channels)
N3	12	0x00001000		0 = OK, 1 = Overload (Bits 0-2 flag the error channels)
	13	0x00002000		0 = OK, 1 = Outside operating conditions
	14	0x00004000		0 = OK, 1 = Startup
	15	0x00008000		0 = OK, 1 = System integrity error
N4	16	0x00010000	Temperature of the X gyro sensor 0°C = 0x0000 1 LSB = 2^{-8} °C	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

Table 47: µIMU IMU Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Reset Acknowledged	0 = Normal, 1 = Reset
	1	0x00000002	Gyros Not Initialized	0 = Normal, 1 = Not Initialized
	2	0x00000004	Gyro X Warning	0 = Normal, 1 = Warning
	3	0x00000008	Gyro Y Warning	0 = Normal, 1 = Warning
N1	4	0x00000010	Gyro Z Warning	0 = Normal, 1 = Warning
	5	0x00000020	Gyro X NOGO	0 = Normal, 1 = NOGO
	6	0x00000040	Gyro Y NOGO	0 = Normal, 1 = NOGO
	7	0x00000080	Gyro Z NOGO	0 = Normal, 1 = NOGO
N2	8	0x00000100	Reserved	
	9	0x00000200	Accels Not Initialized	0 = Normal, 1 = Not Initialized
	10	0x00000400	Accel X Warning	0 = Normal, 1 = Warning
	11	0x00000800	Accel Y Warning	0 = Normal, 1 = Warning
N3	12	0x00001000	Accel Z Warning	0 = Normal, 1 = Warning
	13	0x00002000	Accel X NOGO	0 = Normal, 1 = NOGO
	14	0x00004000	Accel Y NOGO	0 = Normal, 1 = NOGO
	15	0x00008000	Accel Z NOGO	0 = Normal, 1 = NOGO
N4	16	0x00010000	IMU temperature reading as follows: Signed 2-byte value (Short) 1 LSB = 3.90625e^-3 °C Temperature Range +/- 128 °C	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

5.2.42 RAWIMUS Short Raw IMU Data

This log is the short header version of the *RAWIMU* log (page 166).

Message ID: 325

Log Type: Asynch

Recommended Input:

```
log rawimusa onnew
```

ASCII Example:

```
%RAWIMUSA,1105,425384.180;1105,425384.156166800,111607,43088060,430312,
-3033352,-132863,186983,823*5aa97065
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GNSS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	IMU Status	The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs. For the raw IMU status, see one of the following tables: • *Table 36, iIMU-FSAS IMU Status* on page 168 • *Table 37, Litef LCI-1 IMU Status* on page 169 • *Table 39, HG1700 IMU Status* on page 170 • *Table 40, LN200 IMU Status* on page 171 • *Table 41, ISA-100 and ISA-100C IMU Status* on page 172 • *Table 42, IMU-CPT and SPAN-CPT IMU Status* on page 173 • *Table 43, IMU-KVH1750 and IMU-KVH1725 IMU Status* on page 174 • *Table 44, HG1900 and HG1930 IMU Status* on page 175 • *Table 45, ADIS16488, IMU-IGM-A1 and SPAN-IGM-A1 IMU Status* on page 176 • *Table 46, STIM300, IMU-IGM-S1, SPAN-IGM-S1 IMU Status* on page 177 • *Table 47, µIMU IMU Status* on page 178 Also refer to Interface Control Documentation as provided by Honeywell or Northrop Grumman.	Hex Ulong	4	H+12
5	Z Accel Output	Change in velocity count along z axis [a]	Long	4	H+16
6	- (Y Accel Output)	- (Change in velocity count along y axis) [a, b]	Long	4	H+20
7	X Accel Output	Change in velocity count along x axis [a]	Long	4	H+24

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
8	Z Gyro Output	Change in angle count around z axis [c] Right-handed	Long	4	H+28
9	- (Y Gyro Output)	- (Change in angle count around y axis) [b, c] Right-handed	Long	4	H+32
10	X Gyro Output	Change in angle count around x axis [c] Right-handed	Long	4	H+36
11	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. The change in velocity (acceleration) scale factor for each IMU type can be found in *Table 48, Raw IMU Scale Factors* on page 180. Multiply the scale factor in *Table 48, Raw IMU Scale Factors* by the count in this field for the velocity increments.

b. A negative value implies the output is along the positive y-axis marked on the IMU.
A positive value implies the change is in the direction opposite to that of the y-axis marked on the IMU.

c. The change in angle (gyro) scale factor can be found in *Table 48, Raw IMU Scale Factors* on page 180. Multiply the appropriate scale factor in *Table 48, Raw IMU Scale Factors* by the count in this field for the angle increments in radians. To obtain acceleration in m/s^2, multiply the velocity increments by the output rate of the IMU (e.g., 100 Hz for HG1700, HG1900 and HG1930; 200 Hz for ISA-100, ISA-100C, iMAR-FSAS, LN200, LCI-1, µIMU, KVH1750 and ADIS16488; 125 Hz for STIM300).

Table 48: Raw IMU Scale Factors

	Gyroscope Scale Factor	Acceleration Scale Factor
HG1700-AG11 HG1700-AG58 HG1900-CA29/CA50 HG1930-AA99/CA50	2.0^{-33} rad/LSB	2.0^{-27} ft/s/LSB
HG1700-AG17 HG1700-AG62	2.0^{-33} rad/LSB	2.0^{-26} ft/s/LSB
IMU-CPT IMU-KVH1750 SPAN-CPT	0.1 / (3600.0x256.0) rad/LSB	$0.05/2^{15}$ m/s/LSB
iIMU-FSAS	0.1×2^{-8} arcsec/LSB	0.05×2^{-15} m/s/LSB
Litef LCI-1	4×2^{-31} deg/LSB	2×2^{-31} m/s/LSB
LN-200	2^{-19} rad/LSB	2^{-14} m/s/LSB
ISA-100 ISA-100C µIMU	$1.0e^{-9}$ rad/LSB	$2.0e^{-8}$ m/s/LSB
ADIS16488 IMU-IGM-A1 SPAN-IGM-A1	$720/2^{31}$ deg/LSB	$200/2^{31}$ m/s/LSB
STIM300 IMU-IGM-S1 SPAN-IGM-S1	2^{-21} deg/LSB	2^{-22} m/s/LSB

5.2.43 RAWIMUSX IMU Data Extended

This is the short header version of the extended RAWIMUX log intended for use with post-processing. The extended version includes IMU information that is used by the NovAtel Inertial Explorer post-processing software.

Message ID: 1462

Log Type: Asynch

Recommended Input:

```
log rawimusxb onnew
```

ASCII example:

```
%RAWIMUSXA,1692,484620.664;00,11,1692,484620.664389000,00801503,43110635,-817242,
-202184,-215194,-41188,-9895*a5db8c7b
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header (short)	-	H	0
2	IMU Error	Simple IMU error flag. 01 = IMU error 00 = IMU okay. If there is an IMU error, check the IMU Status field for details. This field is output as a Hex value.	Uchar	1	H
3	IMU Type	IMU Type identifier. See *Table 14, IMU Type* on page 39.	Uchar	1	H+1
4	GNSS Week	GNSS Week	Ushort	2	H+2
5	GNSS Week Seconds	Seconds from week start	Double	8	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	IMU Status	The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs. For the raw IMU status, see one of the following tables: • *Table 36, iIMU-FSAS IMU Status* on page 168 • *Table 37, Litef LCI-1 IMU Status* on page 169 • *Table 39, HG1700 IMU Status* on page 170 • *Table 40, LN200 IMU Status* on page 171 • *Table 41, ISA-100 and ISA-100C IMU Status* on page 172 • *Table 42, IMU-CPT and SPAN-CPT IMU Status* on page 173 • *Table 43, IMU-KVH1750 and IMU-KVH1725 IMU Status* on page 174 • *Table 44, HG1900 and HG1930 IMU Status* on page 175 • *Table 45, ADIS16488, IMU-IGM-A1 and SPAN-IGM-A1 IMU Status* on page 176 • *Table 46, STIM300, IMU-IGM-S1, SPAN-IGM-S1 IMU Status* on page 177 • *Table 47, μIMU IMU Status* on page 178 Also refer to Interface Control Documentation as provided by Honeywell or Northrop Grumman.	Hex Ulong	4	H+12
7	Z Accel	Change in velocity count along Z-axis. [a]	Long	4	H+16
8	-(Y Accel)	- (Change in velocity count along y-axis.) [a, b]	Long	4	H+20
9	X Accel	Change in velocity count along x axis. [a]	Long	4	H+24
10	Z Gyro	Change in angle count around z axis. [c] Right-handed	Long	4	H+28
11	-(Y Gyro)	- (Change in angle count around y axis.) [b, c] Right-handed	Long	4	H+32
12	X Gyro	Change in angle count around x axis. [c] Right-handed	Long	4	H+36
13	XXXX	32-bit CRC (ASCII, Binary, and Short Binary only)	Hex	4	H+40
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. The change in velocity (acceleration) scale factor for each IMU type can be found in *Table 48, Raw IMU Scale Factors* on page 180. Multiply the scale factor in *Table 48, Raw IMU Scale Factors* by the count in this field for the velocity increments.

b. A negative value implies the output is along the positive y-axis marked on the IMU.
A positive value implies the change is in the direction opposite to that of the y-axis marked on the IMU.

c. The change in angle (gyro) scale factor can be found in *Table 48, Raw IMU Scale Factors* on page 180. Multiply the appropriate scale factor in *Table 48, Raw IMU Scale Factors* by the count in this field for the angle increments in radians. To obtain acceleration in m/s^2, multiply the velocity increments by the output rate of the IMU (e.g., 100 Hz for HG1700, HG1900 and HG1930; 200 Hz ISA-100, ISA-100C, iMAR-FSAS, LN200, LCI-1, μIMU, KVH1750 and ADIS16488; 125 Hz for STIM300).

5.2.44 RAWIMUX IMU Data Extended

This log is an extended version of the RAWIMU log intended for use with post-processing. The extended version includes IMU information that is used by the NovAtel Inertial Explorer post-processing software.

Message ID: 1461

Log Type: Asynch

Recommended Input:

```
log rawimuxb onnew
```

ASCII example:

```
#RAWIMUXA,COM1,0,81.5,FINESTEERING,1691,410338.819,004c0020,3fd1,43495;00,5,1691,
410338.818721000,00170705,-113836,-464281,43146813,89,11346,181*01cd06bf
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	IMU Error	Simple IMU error flag. 01 = IMU error 00 = IMU okay. If there is an IMU error, check the IMU Status field for details. This field is output as a Hex value.	Uchar	1	H
3	IMU Type	IMU Type identifier. See *Table 14, IMU Type* on page 39.	Uchar	1	H+1
4	GNSS Week	GNSS Week	Ushort	2	H+2
5	GNSS Week Seconds	Seconds from week start	Double	8	H+4

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	IMU Status	The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs. For the raw IMU status, see one of the following tables: • *Table 36, iIMU-FSAS IMU Status* on page 168 • *Table 37, Litef LCI-1 IMU Status* on page 169 • *Table 39, HG1700 IMU Status* on page 170 • *Table 40, LN200 IMU Status* on page 171 • *Table 41, ISA-100 and ISA-100C IMU Status* on page 172 • *Table 42, IMU-CPT and SPAN-CPT IMU Status* on page 173 • *Table 43, IMU-KVH1750 and IMU-KVH1725 IMU Status* on page 174 • *Table 44, HG1900 and HG1930 IMU Status* on page 175 • *Table 45, ADIS16488, IMU-IGM-A1 and SPAN-IGM-A1 IMU Status* on page 176 • *Table 46, STIM300, IMU-IGM-S1, SPAN-IGM-S1 IMU Status* on page 177 • *Table 47, µIMU IMU Status* on page 178 Also refer to Interface Control Documentation as provided by Honeywell or Northrop Grumman.	Hex Ulong	4	H+12
7	Z Accel	Change in velocity count along Z-axis. [a]	Long	4	H+16
8	-(Y Accel)	- (Change in velocity count along y-axis.) [a, b]	Long	4	H+20
9	X Accel	Change in velocity count along x axis. [a]	Long	4	H+24
10	Z Gyro	Change in angle count around z axis. [c] Right-handed	Long	4	H+28
11	-(Y Gyro)	- (Change in angle count around y axis.) [b, c] Right-handed	Long	4	H+32
12	X Gyro	Change in angle count around x axis. [c] Right-handed	Long	4	H+36
13	XXXX	32-bit CRC (ASCII, Binary, and Short Binary only)	Hex	4	H+40
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. The change in velocity (acceleration) scale factor for each IMU type can be found in *Table 48, Raw IMU Scale Factors* on page 180. Multiply the scale factor in *Table 48, Raw IMU Scale Factors* by the count in this field for the velocity increments.

b. A negative value implies the output is along the positive y-axis marked on the IMU.
A positive value implies the change is in the direction opposite to that of the y-axis marked on the IMU.

c. The change in angle (gyro) scale factor can be found in *Table 48, Raw IMU Scale Factors* on page 180. Multiply the appropriate scale factor in *Table 48, Raw IMU Scale Factors*, by the count in this field for the angle increments in radians. To obtain acceleration in m/s^2, multiply the velocity increments by the output rate of the IMU (e.g., 100 Hz for HG1700, HG1900 and HG1930; 200 Hz ISA-100, ISA-100C, iMAR-FSAS, LN200, LCI-1, µIMU KVH1750 and ADIS16488; 125 Hz for STIM300).

5.2.45 RELINSPVA Relative INSPVA log

This log provides the relative offset between the Master and Rover Inertial Solutions. The output solution provides the offset of where the local station is with respect to the other station.

Message ID: 1446

Log Type: Asynch

Recommended Input:

```
LOG RELINSPVAA ONNEW
```

ASCII example:

```
#RELINSPVAA,COM1,0,61.0,FINESTEERING,1805,245074.000,00000000,2338,45757;BODY,
9.285958662,-0.755483058,0.079229338,0.001739020,-0.000126304,0.001525848,0.321033045,
0.669367786,4.466250181,0.000000000,"b81V",INS_ALIGNMENT_COMPLETE,"B20C",
INS_ALIGNMENT_COMPLETE,NARROW_INT,00000000*a114ce3c
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Output Frame	The current output frame (IMU body, ECEF or local level frame). The output frame is specified using the SETRELINSOUTPUTFRAME command. See *Section 4.2.46, SETRELINSOUTPUTFRAME* on page 94	Enum	4	H
3	DeltaPosX	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the X-axis ECEF = along the X-axis Local level = Northing	Double	8	H+4
4	DeltaPosY	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the Y-axis ECEF = along the Y-axis Local level = Easting	Double	8	H+12
5	DeltaPosZ	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the Z-axis ECEF = along the Z-axis Local level = Up	Double	8	H+20
6	DeltaVelX	Difference in velocity between the two receivers (m/s). The position difference is relative to the output frame: BODY = along the X-axis ECEF = along the X-axis Local level = Northing	Double	8	H+28

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
7	DeltaVelY	Difference in velocity between two receivers (m/s). The position difference is relative to the output frame: BODY = along the Y-axis ECEF = along the Y-axis Local level = Easting	Double	8	H+36
8	DeltaVelZ	Difference in velocity between the two receivers (m/s). The position difference is relative to the output frame: BODY = along the Z-axis ECEF = along the Z-axis Local level = Up	Double	8	H+44
9	DeltaRoll	Difference in roll between the two receivers (degrees).	Double	8	H+52
10	DeltaPitch	Difference in pitch between the two receivers (degrees).	Double	8	H+60
11	DeltaHeading	Difference in heading between the two receivers (degrees).	Double	8	H+68
12	Diff Age	Differential age in seconds.	Float	4	H+76
13	Rover ID	Rover receiver ID string.	Char[4]	4	H+80
14	Rover INSStatus	INS status of the rover receiver. See *Table 32, Inertial Solution Status* on page 136	Enum	4	H+84
15	Master ID	Master receiver ID string.	Char[4]	4	H+88
16	Master INSStatus	INS status of the master receiver. See *Table 32, Inertial Solution Status* on page 136	Enum	4	H+92
17	RTK Status	Status of the current RTK vector between master and rover. See *Table 25, Position or Velocity Type* on page 114	Enum	4	H+96
18	ExtStatus	Extended solution status.	Hex	4	H+100
20	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+104
21	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.46 SYNCHEAVE
Synchronous Log Containing the Instantaneous Heave Value

Synchronous heave is available up to the rate of the IMU. It can also be logged 'on time' at lower rates.

This log also includes information about the estimated accuracy of the heave value through the standard deviation of the heave.

You must have an inertial solution to use this log. The heave filter must be enabled using the `HEAVEFILTER` command, see page 52, before this log is available.

Message ID: 1708

Log Type: Synch

Recommended Input:

```
log syncheavea ontime 0.05
```

ASCII example:

```
#SYNCHEAVEA,COM1,0,50.0,FINESTEERING,1770,245720.925,00000020,552e,12622;
-0.045410579,0.436800622*b8c14286
```

Field #	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	Heave	Instantaneous heave value (metres)	Double	8	H
3	Std. Dev.	Standard deviation of the heave value (metres)	Double	8	H+8
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+16
5	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.47 SYNCRELINSPVA
Synchronous Relative INSPVA log

This log provides the relative offset between the master and rover inertial solutions. The output solution provides the offset of where the local station is with respect to the other station.

> This log is designed to provide synchronous, relative Position, Velocity and Attitude information, propagating the information between matched corrections between the master and remote solutions. It is highly recommended that the highest rate of corrections be used at all times for the most precise and robust performance.

Message ID: **1743**

Log Type: **Synch**

Recommended Input:

```
LOG SYNCRELINSPVAA ONTIME 1
```

ASCII example:

```
#SYNCRELINSPVAA,COM1,0,72.5,FINESTEERING,1805,247243.000,00000000,e9c7,13005;BODY,
8.141080733,-2.779177478,2.045421773,-0.001464009,-0.001038329,0.002323548,0.409467974,
0.715633909,-6.204731538,0.000000000,"B81V",INS_ALIGNMENT_COMPLETE,"B20C",
INS_ALIGNMENT_COMPLETE,INS_PSRSP,00000000*e270f5c8
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Output Frame	The current output frame (IMU body, ECEF or local level frame). The output frame is specified using the SETRELINSOUTPUTFRAME command. See *Section 4.2.46, SETRELINSOUTPUTFRAME* on page 94	Enum	4	H
3	DeltaPosX	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the X-axis ECEF = along the X-axis Local level = Northing	Double	8	H+4
4	DeltaPosY	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the Y-axis ECEF = along the Y-axis Local level = Easting	Double	8	H+12
5	DeltaPosZ	Difference in the position between the two receivers (m). The position difference is relative to the output frame: BODY = along the Z-axis ECEF = along the Z-axis Local level = Up	Double	8	H+20

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
6	DeltaVelX	Difference in velocity between the two receivers (m/s). The position difference is relative to the output frame: BODY = along the X-axis ECEF = along the X-axis Local level = Northing	Double	8	H+28
7	DeltaVelY	Difference in velocity between two receivers (m/s). The position difference is relative to the output frame: BODY = along the Y-axis ECEF = along the Y-axis Local level = Easting	Double	8	H+36
8	DeltaVelZ	Difference in velocity between the two receivers (m/s). The position difference is relative to the output frame: BODY = along the Z-axis ECEF = along the Z-axis Local level = Up	Double	8	H+44
9	DeltaRoll	Difference in roll between the two receivers (degrees).	Double	8	H+52
10	DeltaPitch	Difference in pitch between the two receivers (degrees).	Double	8	H+60
11	DeltaHeading	Difference in heading between the two receivers (degrees).	Double	8	H+68
12	Diff Age	Differential age in seconds.	Float	4	H+76
13	Rover ID	Rover receiver ID string.	Char[4]	4	H+80
14	Rover INSStatus	INS status of the rover receiver. See *Table 32, Inertial Solution Status* on page 136	Enum	4	H+84
15	Master ID	Master receiver ID string.	Char[4]	4	H+88
16	Master INSStatus	INS status of the master receiver. See *Table 32, Inertial Solution Status* on page 136	Enum	4	H+92
17	RTK Status	Status of the current RTK vector between master and rover. See *Table 25, Position or Velocity Type* on page 114	Enum	4	H+96
18	ExtStatus	Extended solution status.	Hex	4	H+100
20	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+104
21	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.48 TAGGEDMARK1PVA Position, Velocity and Attitude at a Tagged Mark1

> TAGGEDMARK1PVA contains the same information as MARK1PVA with the addition of a unique identifying number (tag).

The user specifies a TAG for the upcoming TAGGEDMARKPVA via the `TAGNEXTMARK` command. That tag shows up at the end of this message, which is otherwise identical to the MARK1PVA message.

Message ID: 1258

Log Type: Synch

Recommended Input:

```
log taggedmark1pva onnew
```

Abbreviated ASCII Example:

```
#TAGGEDMARK1PVAA,COM1,0,63.0,FINESTEERING,1732,247787.965,004c0020,ae1e,12002;
1732,247787.964913500,51.11693231436,-114.03884974751,1046.9481,0.0001,0.0007,
0.0004,1.090392628,0.766828598,244.413950146,INS_SOLUTION_GOOD,1234*34fda4f4
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	Week	GNSS Week at Mark 1 request	Ulong	4	H
3	Seconds into Week	GNSS Seconds at Mark1 request	Double	8	H+4
4	Latitude	Latitude at Mark 1 request	Double	8	H+12
5	Longitude	Longitude at Mark 1 request	Double	8	H+20
6	Height	Height at Mark 1 request	Double	8	H+28
7	North Velocity	North Velocity at Mark 1 request	Double	8	H+36
8	East Velocity	East Velocity at Mark1 request	Double	8	H+44
9	Up Velocity	Up Velocity at Mark 1 request	Double	8	H+52
10	Roll	Roll at Mark1 request	Double	8	H+60
11	Pitch	Pitch at Mark1 request	Double	8	H+68
12	Azimuth	Azimuth at Mark1 request	Double	8	H+76
13	Status	INS Status at Mark 1 request	Enum	4	H+84
14	Tag	Tag ID from `TAGNEXTMARK` command, if any (default = 0)	Ulong	4	H+88
15	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+92
16	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.49 TAGGEDMARK2PVA Position, Velocity and Attitude at a Tagged Mark2

TAGGEDMARK2PVA contains the same information as MARK2PVA with the addition of a unique identifying number (tag).

The SPAN-CPT does not have a Mark2 input, so this log is not available for the SPAN-CPT.

The user specifies a TAG for the upcoming TAGGEDMARKPVA via the `TAGNEXTMARK` command. That tag shows up at the end of this message, which is otherwise identical to the MARK2PVA message.

Message ID: 1259

Log Type: Synch

Recommended Input:

```
log taggedmark2pva onnew
```

Abbreviated ASCII Example:

```
#TAGGEDMARK2PVAA,COM1,0,73.0,FINESTEERING,1732,248347.693,000500a0,2ab3,12002;
1732,248347.692695400,51.11693017508,-114.03884746120,1046.3929,0.0009,0.0014,
0.0015,0.559580646,1.121028629,255.541153133,INS_SOLUTION_GOOD,1234*1e97dd88
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	Week	GNSS Week at Mark2 request	Ulong	4	H
3	Seconds into Week	GNSS Seconds at Mark2 request	Double	8	H+4
4	Latitude	Latitude at Mark2 request	Double	8	H+12
5	Longitude	Longitude at Mark2 request	Double	8	H+20
6	Height	Height at Mark2 request	Double	8	H+28
7	North Velocity	North Velocity at Mark2 request	Double	8	H+36
8	East Velocity	East Velocity at Mark2 request	Double	8	H+44
9	Up Velocity	Up Velocity at Mark2 request	Double	8	H+52
10	Roll	Roll at Mark2 request	Double	8	H+60
11	Pitch	Pitch at Mark2 request	Double	8	H+68
12	Azimuth	Azimuth at Mark2 request	Double	8	H+76
13	Status	INS Status at Mark2 request	Enum	4	H+84
14	Tag	Tag ID from TAGNEXTMARK command, if any (default = 0)	Ulong	4	H+88
15	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+92
16	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.50 TAGGEDMARK3PVA Position, Velocity and Attitude at a Tagged Mark3

TAGGEDMARK3PVA contains the same information as MARK3PVA with the addition of a unique identifying number (tag).

This log is available only on SPAN systems with an OEM638 or ProPak6 receiver.

The user specifies a TAG for the upcoming TAGGEDMARKPVA via the `TAGNEXTMARK` command. That tag shows up at the end of this message, which is otherwise identical to the MARK3PVA message.

Message ID: 1327

Log Type: Synch

Recommended Input:

```
log taggedmark3pva onnew
```

Abbreviated ASCII Example:

```
#TAGGEDMARK3PVAA,COM1,0,73.0,FINESTEERING,1732,248347.693,000500a0,2ab3,12002;
1732,248347.692695400,51.11693017508,-114.03884746120,1046.3929,0.0009,0.0014,
0.0015,0.559580646,1.121028629,255.541153133,INS_SOLUTION_GOOD,1234*1e97dd88
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	Week	GNSS Week at Mark3 request	Ulong	4	H
3	Seconds into Week	GNSS Seconds at Mark3 request	Double	8	H+4
4	Latitude	Latitude at Mark3 request	Double	8	H+12
5	Longitude	Longitude at Mark3 request	Double	8	H+20
6	Height	Height at Mark3 request	Double	8	H+28
7	North Velocity	North Velocity at Mark3 request	Double	8	H+36
8	East Velocity	East Velocity at Mark3 request	Double	8	H+44
9	Up Velocity	Up Velocity at Mark3 request	Double	8	H+52
10	Roll	Roll at Mark3 request	Double	8	H+60
11	Pitch	Pitch at Mark3 request	Double	8	H+68
12	Azimuth	Azimuth at Mark3 request	Double	8	H+76
13	Status	INS Status at Mark3 request	Enum	4	H+84
14	Tag	Tag ID from TAGNEXTMARK command, if any (default = 0)	Ulong	4	H+88
15	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+92
16	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.51 TAGGEDMARK4PVA
Position, Velocity and Attitude at a Tagged Mark4

TAGGEDMARK4PVA contains the same information as MARK4PVA with the addition of a unique identifying number (tag).

This log is available only on SPAN systems with an OEM638 or ProPak6 receiver.

The user specifies a TAG for the upcoming TAGGEDMARKPVA via the `TAGNEXTMARK` command. That tag shows up at the end of this message, which is otherwise identical to the MARK4PVA message.

Message ID: 1328

Log Type: Synch

Recommended Input:

```
log taggedmark4pva onnew
```

Abbreviated ASCII Example:

```
#TAGGEDMARK4PVAA,COM1,0,73.0,FINESTEERING,1732,248347.693,000500a0,2ab3,12002;
1732,248347.692695400,51.11693017508,-114.03884746120,1046.3929,0.0009,0.0014,
0.0015,0.559580646,1.121028629,255.541153133,INS_SOLUTION_GOOD,1234*1e97dd88
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	Week	GNSS Week at Mark4 request	Ulong	4	H
3	Seconds into Week	GNSS Seconds at Mark4 request	Double	8	H+4
4	Latitude	Latitude at Mark4 request	Double	8	H+12
5	Longitude	Longitude at Mark4 request	Double	8	H+20
6	Height	Height at Mark4 request	Double	8	H+28
7	North Velocity	North Velocity at Mark4 request	Double	8	H+36
8	East Velocity	East Velocity at Mark4 request	Double	8	H+44
9	Up Velocity	Up Velocity at Mark4 request	Double	8	H+52
10	Roll	Roll at Mark4 request	Double	8	H+60
11	Pitch	Pitch at Mark4 request	Double	8	H+68
12	Azimuth	Azimuth at Mark4 request	Double	8	H+76
13	Status	INS Status at Mark4 request	Enum	4	H+84
14	Tag	Tag ID from TAGNEXTMARK command, if any (default = 0)	Ulong	4	H+88
15	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+92
16	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.52 TIMEDWHEELDATA Timed Wheel Data

This log contains time stamped wheel sensor data. The time stamp in the header is the time of validity for the wheel data and not the time the TIMEDWHEELDATA log was output.

See the relevant SPAN User Manual for information about wheel sensor messages.

> Depending on the method used to connect the wheel sensor (through an IMU using a UIC, an IMU in an IMU Enclosure (IMU-ISA-100C, IMU-ISA100, IMU-HG1900, IMU-LN200 or IMU-μIMU), an iIMU-FSAS or an IMU-CPT, or directly into an IMU-IGM enclosure), either field 3 or field 4 of the log will be filled for wheel velocity. They are equivalent, but are filled differently depending on what data is provided to SPAN.
>
> Note that neither velocity value is used by the SPAN filter. Rather, the SPAN filter uses cumulative ticks per second.

Message ID: 622

Log Type: Asynch

Recommended Input:

```
log timedwheeldataa onnew
```

ASCII Example:

```
%TIMEDWHEELDATAA,1393,411345.001;58,0,215.814910889,0,0,1942255*3b5fa236
```

> This example is from the iMAR iMWS wheel sensor.

Field #	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header (short header)	-	H	0
2	Ticks Per Rev	Number of ticks per revolution	Ushort	2	H
3	Wheel Vel	Wheel velocity in counts/s	Ushort	2	H+2
4	fWheel Vel	Float wheel velocity in counts/s	Float	4	H+4
5	Reserved		Ulong	4	H+8
6			Ulong	4	H+12
7	Cumulative Ticks	Number of ticks	Long	4	H+16
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+20
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.53 TSS1
TSS1 Protocol for Heave, Roll and Pitch

This log provides heave, roll and pitch information in TSS1 protocol.

> This message is in a different format than any other log output by the SPAN system.

> To use this log, make sure that
> - Heave is enabled (see *HEAVEFILTER* on page 52)
> - the INS status is greater than INS_ALIGNMENT_COMPLETE

Message ID: 1456

Log Type: Synch

Recommended Input:

```
log tss1a ontime 1
```

Message Format:

```
:XXAAAASMHHHHQMRRRRSMPPPP<CR><LF>
```

ASCII Example:

```
:00FFCA -0003F-0325  0319
```

Field	Field Type	Description	Symbol	Example
1	Log Header	Log header	-	0
2	Horizontal Acceleration	Horizontal acceleration from 0 to 9.81m/s^2. Shown as a one byte unsigned hex number where the least significant bit = 3.83 cm/s^2.	XX	00
3	Vertical Acceleration	Vertical acceleration from -20.48 to +20.48 m/s^2. Shown as a two byte hex number where the least significant bit = 0.0625 cm/s^2.	AAAA	FFCA
4	Space Character	A space delimiter.	S	
5	Heave Polarity	Space if positive. Minus sign (-) if negative.	M	-
6	Heave	Heave value from -99.99 to +99.99 m. Shown as a four digit integer where the least significant bit = 0.01 m.	HHHH	0003
7	Status Flag	F if INS Active. H if INS has not completed an alignment.	Q	F
8	Roll Polarity	Space if positive. Minus sign (-) if negative.	M	-
9	Roll	Roll value from -99.99 to +99.99 degrees. Shown as a four digit integer where the least significant bit = 0.01 degrees.	RRRR	0325
10	Space Character	A space delimiter.	S	

Field	Field Type	Description	Symbol	Example
11	Pitch Polarity	Space if positive. Minus sign (-) if negative.	M	
12	Pitch	Pitch value from -99.99 to +99.99 degrees. Shown as a four digit integer where the least significant bit = 0.01 degrees.	PPPP	0319
13	[CR][LF]	Sentence terminator	<CR><LF>	

5.2.54 VARIABLELEVERARM Display Variable Lever Arm Details

Use this log to redisplay the re-calculated variable lever arm whenever a new INPUTGIMBALANGLE command is received.

> This log is not output until the INS alignment is complete.

Message ID: **1320**

Log Type: **Asynch**

Recommended Input:

```
log variableleverarma onnew
```

ASCII Example:

```
#VARIABLELEVERARMA,SPECIAL,0,81.5,FINESTEERING,1614,495820.512,40040000,0000,320;
-0.0959421909646755,0.1226971902356540,1.1319295452903300,
0.0100057787272846,0.0122604827412661,0.1131929545290330*9611d3c6
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	XOffset	SPAN computational frame x-axis offset	Double	8	H
3	YOffset	SPAN computational frame y-axis offset	Double	8	H+8
4	ZOffset	SPAN computational frame z-axis offset	Double	8	H+16
5	XUncert	X-axis uncertainty in metres	Double	8	H+24
6	YUncert	Y-axis uncertainty in metres	Double	8	H+32
7	ZUncert	Z-axis uncertainty in metres	Double	8	H+40
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+48
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

5.2.55 VEHICLEBODYROTATION Vehicle to SPAN Frame Rotation

This log reports the angular offset from the vehicle frame to the SPAN frame. The SPAN frame is defined by the transformed IMU enclosure axis with z pointing up (refer *SETIMUORIENTATION* on page 74).

> If the IMU is mounted with the z-axis pointing up (as marked on the IMU enclosure), the IMU enclosure frame is the same as the SPAN frame.

This log reports whatever was entered using the `VEHICLEBODYROTATION` command (page 102) or whatever was solved for after using the `RVBCALIBRATE` command (page 70).

Refer to *VEHICLEBODYROTATION* on page 102 for more information.

Message ID: 642

Recommended Input:

```
log vehiclebodyrotationa onchanged
```

ASCII Example:

```
#VEHICLEBODYROTATIONA,COM1,0,36.5,FINESTEERING,1264,144170.094,00000000,bcf2,1541;1.5869
999997474209,2.6639999995760122,77.6649999876392343,2.0000000000000000,2.000000000000000
0,5.0000000000000000*25f886cc
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	X Angle	Right-hand rotation about vehicle frame x-axis in degrees	Double	8	H
3	Y Angle	Right-hand rotation about vehicle frame y-axis in degrees	Double	8	H+8
4	Z Angle	Right-hand rotation about vehicle frame z-axis in degrees	Double	8	H+16
5	X Uncertainty	Uncertainty of x-rotation in degrees (default = 0)	Double	8	H+24
6	Y Uncertainty	Uncertainty of y-rotation in degrees (default = 0)	Double	8	H+32
7	Z Uncertainty	Uncertainty of z-rotation in degrees (default = 0)	Double	8	H+40
8	xxxx	32-bit CRC	Hex	4	H+48
9	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

5.2.56 WHEELSIZE
Wheel Size

This log contains wheel sensor information.

The inertial filter models the size of the wheel to compensate for changes in wheel circumference due to hardware or environmental changes. The default wheel size is 1.96 m. A scale factor to this default size is modeled in the filter and this log contains the current estimate of the wheel size.

Message ID: 646

Log Type: Asynch

Recommended Input:

```
log wheelsizea onnew
```

ASCII Example:

```
#WHEELSIZEA,COM3,0,44.0,FINESTEERING,0,0.000,00000000,85f8,33738;
1.025108123,2.009211922,0.000453791*b65d28e6
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Scale	Wheel sensor scale factor	Double	8	H
3	Circum	Wheel circumference (m)	Double	8	H+8
4	Var	Variance of circumference (m^2)	Double	8	H+16
5	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+24
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Chapter 6 Variable Lever Arm

The variable lever arm concept arose to support applications in which the IMU is no longer rigidly fixed to the vehicle, but rather on a gimballed mount. This creates an issue where the input lever arm offsets to the GNSS antenna are no longer correct, because the IMU can rotate on its mount, while the antenna remains fixed.

The use of the variable lever arm functionality requires that the device to which the IMU is attached be able to send its gimbal rotation angles back to SPAN. These angles are used to recalculate the lever arm at the rate that they are received. SPAN will also be able to output a gimballed solution at the rate the gimbal angles are received.

6.1 Technical Description

There are several frames of reference involved when dealing with a gimballed mount. The frames are all very similar, but can be quite confusing due to small differences. Below are all frames applicable to the implementation of the variable lever arm:

Frame	Description	Notation
SPAN body frame	The physical IMU axes	(b)
SPAN computational frame	Standard SPAN computational frame where Z is up, Y is forward and X is right	(s)
Vehicle frame	SPAN computational frame may be rotated to match the vehicle frame using the `VEHICLEBODYROTATION` command	(v)
Mount body frame	The physical axes of the gimballed mount	(mb)
Gimbal body frame	The physical axes of the gimbal plane that matches the mount body frame when gimbal angles are zero	(gb)
Mount computation frame	Mapping 5 applied to the mount body frame	(mc)
Gimbal computation frame	Mapping 5 applied to the gimbal body frame	(gc)
Mount SPAN frame	Mapping five and `GIMBALSPANROTATION` applied to the mount frame so that the mount and SPAN frames are coincident when gimbal angles are zero	(m)
Gimbal SPAN frame	Mapping 5 and `GIMBALSPANROTATION` applied to the gimbal frame so that the gimbal and SPAN frames are coincident	(g)

Figure 4, Simple Configuration and *Figure 5, Rotation Results* on page 201 illustrate a basic scenario for the information in the table above. *Figure 4, Simple Configuration* shows a possible configuration for the mount body frame and SPAN frame. *Figure 5, Rotation Results* shows the desired result of all rotations: all three frames have the same axis definitions.

Figure 4: Simple Configuration

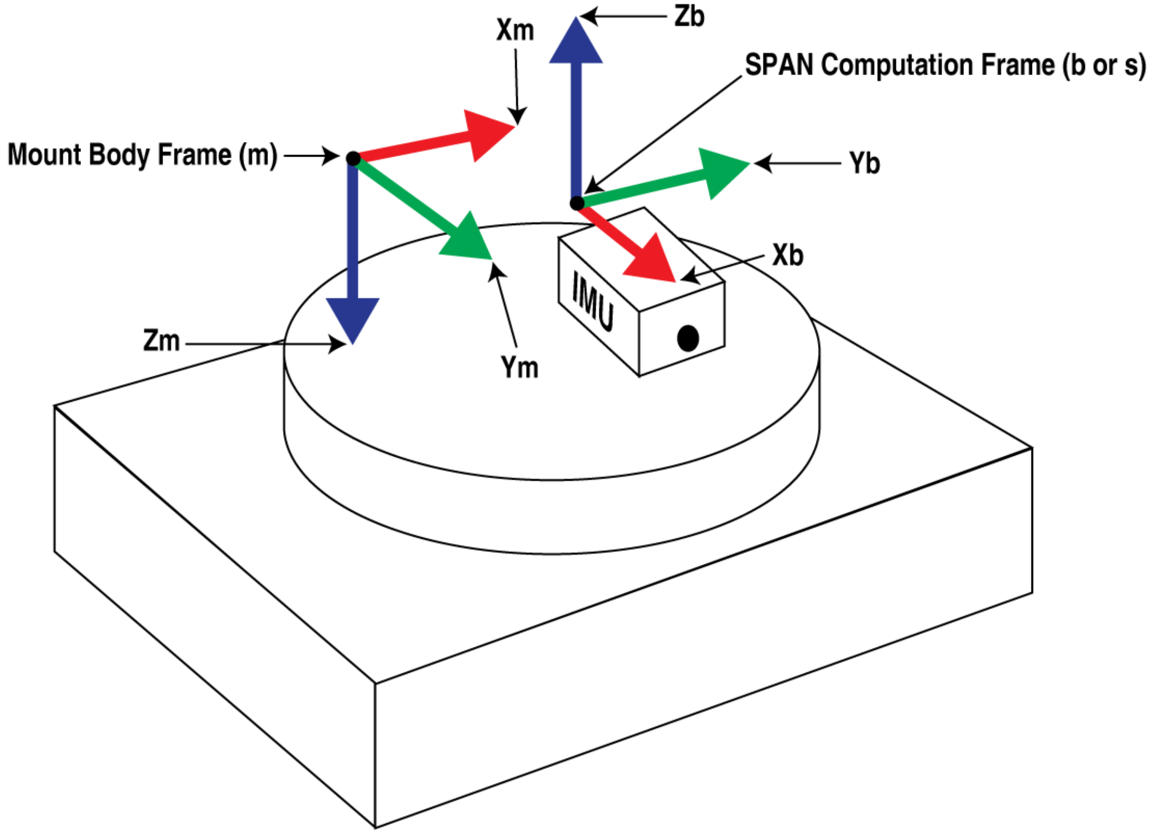
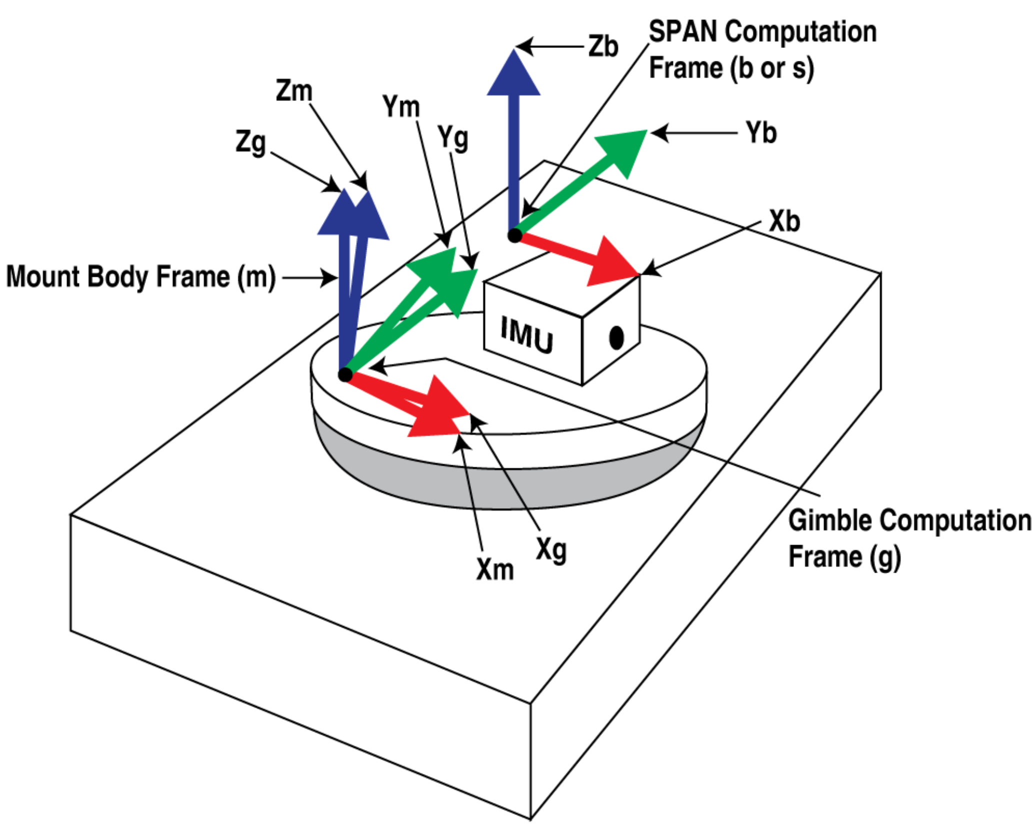


Figure 5: Rotation Results



6.2 How to Use Variable Lever Arm

The variable lever arm functionality is simple to use in a SPAN system. It requires the input of gimbal angles from the camera mount or platform that the IMU is mounted on. After that is provided, the system will automatically compute the variable lever arm and produce several messages for output.

Ensuring a proper variable lever arm is important for SPAN performance.

6.2.1 Basic I/O

The variable lever arm functionality is based on the input of `INPUTGIMBALANGLE` commands. This command inputs the gimbal angles from whatever platform the IMU is mounted on and is input in the mount body frame (mb). Entering this command will automatically cause the system to rotate the static lever arm into the latest gimballed frame. The update rate of the variable lever arm depends on the rate of the gimbal commands which can be entered at 1 to 50 Hz. If an `INPUTGIMBALANGLE` command is not received for over 1 second then the system will return to using the static lever arm.

In addition to using the variable lever arm internally, the following information is available to the user.

Table 49: Logs used with Variable Lever Arm

Log	Description
VARIABLELEVERARM	This log displays the calculated variable lever arm. The format of this log is similar to the BESTLEVERARM logs, except that the mapping field is not displayed. The mapping field is not shown because the VARIABLELEVERARM log is output in SPAN Computation frame rather than the IMU body frame that is used by all the other logs. This makes the mapping field irrelevant for the VARIABLELEVERARM log.
GIMBALLEDPVA	This log has the same format as the INSPVA log, but displays the position, velocity and attitude related back to the camera mount frame. The INSPVA log always displays where the IMU is truly pointing (adjusted by the `VEHICLEBODYROTATION` or `SETINSROTATION` command). When the camera mount is in its zero locked position, the GIMBALLEDPVA log will match the standard INSPVA log.
INSPVA	The INSPVA log is synchronous so does not depend on the entry of INPUTGIMBALANGLE logs. However when the Gimbal angles are non-zero, the INSPVA log no longer represents the orientation of the camera mount / vehicle, but rather the current orientation of the unlocked camera. If the orientation of the camera mount / vehicle are important, use the GIMBALLEDPVA log.

6.2.2 Rotations and Mapping

For the information provided by the system to be meaningful, the rotations have to be set up as desired by the user. By default the system assumes that the IMU and camera mount are oriented to the SPAN standard of Y-forward, Z-up, and X-right. If this is not the case then additional commands are necessary to relate all of the different frames together using the commands below.

Table 50: Commands used with Variable Lever Arm

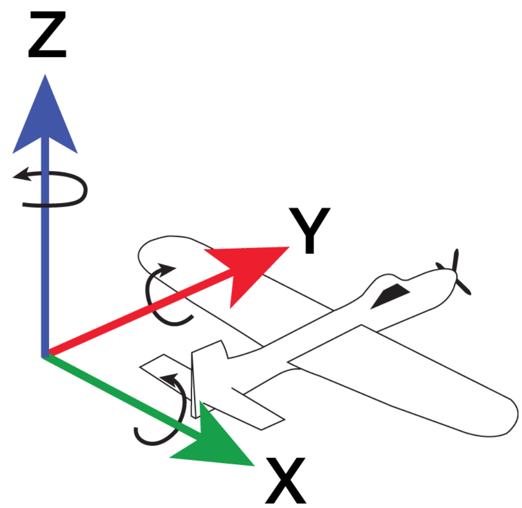
Command	Purpose
SETIMUORIENTATION	The IMU orientation is computed by the SPAN system when it is static using accelerometer data. However, it can also be entered if static data will not be available. The IMU orientation identifies which IMU axis is aligned with gravity and whether it is down or up in relation to gravity. This is required information before the SPAN system can complete an inertial alignment. This command relates the IMU body frame (b) to the SPAN computational frame (c).
SETGIMBALORIENTATION	Similar to the IMU orientation, if the camera mount axes are not in the default orientation, the axis aligned with gravity must be identified. Unlike the IMU orientation, this cannot be determined automatically by the system. This command relates the mount body frame (mb) to the mount computational frame (mc).
VEHICLEBODYROTATION	The default SPAN orientation assumes that the Y-axis of the IMU is mounted forward aligned with the forward axis of the vehicle. If this is not true then this command can be entered to rotate the output so that this is true. This command is applied after the raw data is re-mapped according to the `SETIMUORIENTATION` command, so usually only a rotation about Z is required. For this to take effect the `APPLYVEHICLEBODYROTATION` command must also be sent. This command relates the SPAN computation frame (c) to the vehicle frame (v).
GIMBALSPANROTATION	Similar to the VEHICLEBODYROTATION command, the system must know if the mapped axes of the SPAN system and mount do not match. For example, if the X-axis of the camera mount is aligned with the Y-axis of the SPAN system. This command relates the mount computational frame (mc) to the SPAN computational frame (c).

6.3 The Vehicle Frame

The definition of the vehicle frame is as follows:

- z-axis – points up through the roof of the vehicle perpendicular to the ground
- y-axis – points out the front of the vehicle in the direction of travel
- x-axis – completes the right-handed system (out the right-hand side of the vehicle when facing forward)

Figure 6: Vehicle Frame

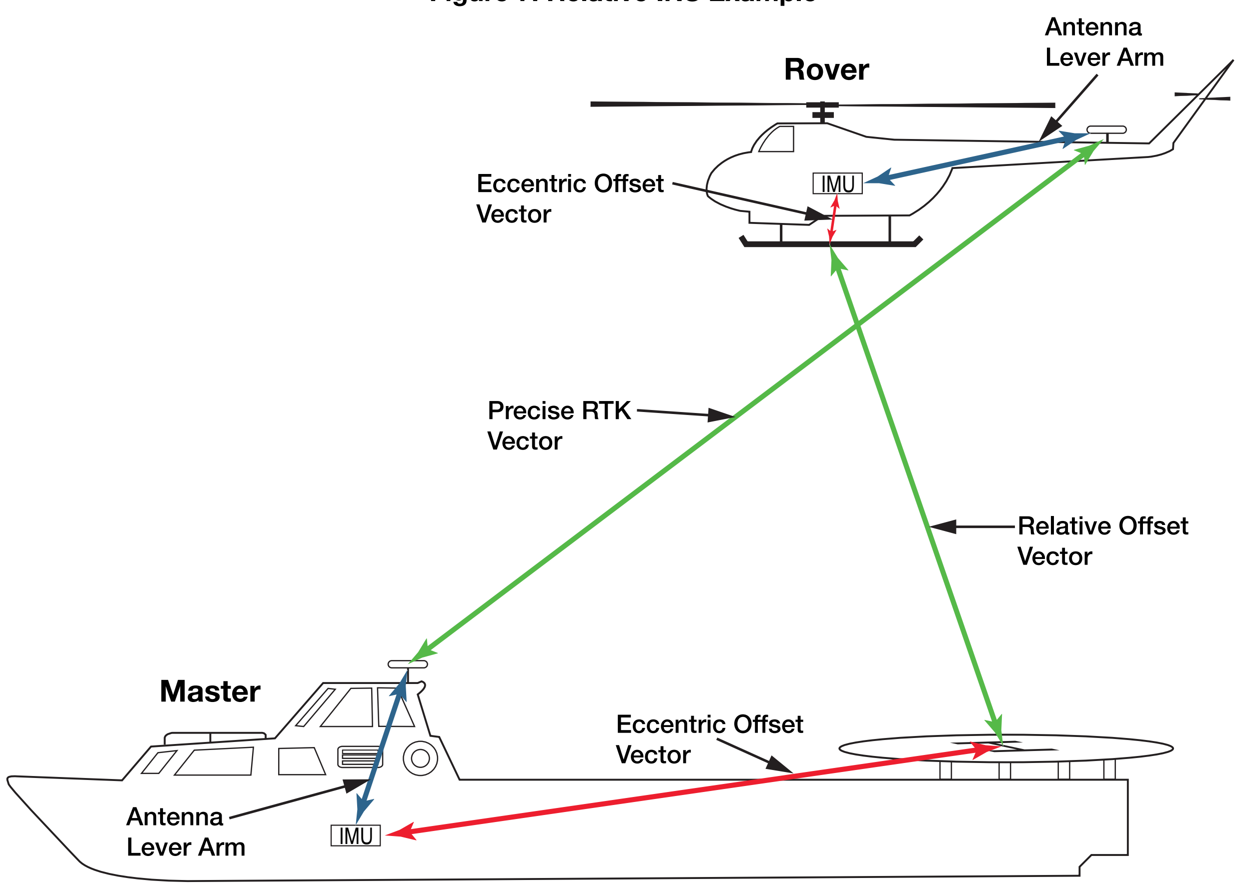


Chapter 7 Relative INS

Novatel's Relative INS technology generates a position, velocity and full attitude vector between two SPAN systems. One is the Master receiver and the other is the Rover receiver. Once configured, the Master receiver begins transmitting corrections to the Rover receiver. Relative information is created and the system begins filling the RELINSPVA and SYNCRELINSPVA logs on the Rover receiver. The RELINSPVA log is then transmitted back to the Master receiver for output.

The data link required must be able to support [864bits+108bits x #obs tracked per second]. The RELINSPVA log is requested ONNEW (available from 1Hz-20Hz depending on the setup of the RELINSAUTOMATION command) while the SYNCRELINSPVA log can be requested ONTIME at up to 200Hz (at the rover only). Position and velocity solutions are differenced in the ECEF frame and then rotated into the local level. The Relative offset vector that is output is dependent on Master/Rover's precise RTK vector and their eccentric offset vectors (optional offset provided by the SETINSOFFSET command), shown in *Figure 7, Relative INS Example*.

Figure 7: Relative INS Example



An important command that can be used with Relative INS to manually change the maximum amount of time to use RTK data is RTKTIMEOUT. This command is used to set the maximum age of RTK data to use when operating as a rover station. RTK data received that is older than the specified time is ignored. The default is 60 seconds, however when using Relative INS, it is suggested this be set to 5 seconds by sending the command RTKTIMEOUT 5.

For information about the RTKTIMEOUT command, refer to the OEM6 Family Firmware Reference Manual (OM-20000129).

7.1 Configure Relative INS

To configure the receivers to begin computing the relative information:

1. Setup a link between the receivers that will be used to transfer data (for example, radios).
2. Enable the transfer of relative corrections between receivers using one of the methods below.
   - Send the `RELINSAUTOMATION` command only at the Rover receiver
   - Send the `RELINSCONFIG` command at both the Master and Rover receiver

> The Model option must be present on both the rover and master for the Relative INS feature to be enabled.