OEM6® Family Firmware Reference Manual

OM-20000129 Rev 8 January 2015

OEM6 Family of Receivers - Firmware Reference Manual

Publication Number:	OM-20000129
Revision Level:	8
Revision Date:	January 2015

This manual reflects firmware version 6.60 / OEM060600RN0000

Proprietary Notice

Information in this document is subject to change without notice and does not represent a commitment on the part of NovAtel Inc. The information contained within this manual is believed to be true and correct at the time of publication.

OEM6, RT-2, ALIGN and NovAtel are registered trademarks of NovAtel Inc.

OEM615, OEM617, OEM617D, OEM628, OEM638, FlexPak6, FlexPak6D, NovAtel CORRECT and GLIDE are trademarks of NovAtel Inc.

All other brand names are trademarks of their respective holders.

Manufactured and protected under U.S. patents:

#5,101,416	#6,184,822 B1
#5,390,207	#6,243,409 B1
#5,414,729	#6,445,354 B1
#5,495,499	#6,608,998 B1
#5,736,961	#6,664,923 B1
#5,809,064	#7,738,536

Table of Contents

4 Responses

Tables

Figures

Customer Support

NovAtel Knowledge Base

If you have a technical issue, visit the NovAtel Support page at www.novatel.com/support.Through the *Support* page, you can contact Customer Support, find papers and tutorials or download current manuals and the latest firmware.

Before Contacting Customer Support

Before you contact 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
    RANGEB ontime 1
    BESTPOSB ontime 1
    RXCONFIGA once
    VERSIONA once
    ```

2. Send the data file to NovAtel Customer Support: support@novatel.com

3. You can also issue a FRESET command to the receiver to clear any unknown settings.

> ⓘ The FRESET command will erase all user settings. You should know your configuration (by requesting the RXCONFIGA log) and be able to reconfigure the receiver before you send the FRESET command.

If you are having a hardware problem, send a list of the troubleshooting steps taken and the results.

Contact Information

Use one of the following methods to contact NovAtel Customer Support:

Call the NovAtel Hotline at 1-800-NOVATEL (U.S. and Canada)	
or +1-403-295-4900 (international) (business hours 9 a.m. to 5 p.m. mountain time)	
Fax: +1-403-295-4901 E-mail: support@novatel.com Website: www.novatel.com	Write: NovAtel Inc. Customer Support Department 1120 - 68 Avenue NE Calgary, AB Canada, T2E 8S5

Foreword

About this Manual

Thank you for purchasing your NovAtel product. Your receiver includes companion documents to this manual with information on the hardware operation. Afterwards, this document will be your primary reference guide for commands and logs.

This manual describes each command and log the OEM6 family of receivers are capable of accepting or generating. Sufficient detail is provided so you can understand the purpose, syntax and structure of each command or log. You will also be able to communicate with the receiver, enabling you to effectively use and write custom interfacing software for specific applications.

Related Documents and Information

OEM6 products include the following:

- Satellite Based Augmentation System (SBAS) signal functionality
- Support for all current and upcoming GNSS constellations
- L-Band capability including TerraStar licensed based corrections
- National Marine Electronics Association (NMEA) standards, a protocol used by GNSS receivers to transmit data
- Differential Global Positioning System (DGPS)
- Real-Time Kinematic (RTK)

For more information on these components, please refer the Support page on our website at www.novatel.com/support. For introductory information on GNSS technology, refer to our *An Introduction to GNSS* book found at www.novatel.com/an-introduction-to-gnss/

This manual does not address any of the receiver hardware attributes or installation information. Consult the OEM6 Family Installation and Operation Manual (OM-20000128) for information about these topics. Furthermore, should you encounter any functional, operational or interfacing difficulties with the receiver, refer to the NovAtel web site for warranty and support information.

Conventions

The following conventions are used in this manual:

ⓘ	Denotes information to supplement or clarify the accompanying text.
▼	Caution that a certain action, operation or configuration may result in incorrect or improper use of the product.
⬢	Warning that a certain action, operation or configuration may result in regulatory noncompliance, safety issues or equipment damage.

Logs and Commands Defaults and Structure

- The factory defaults for commands are shown in *Section 2.3, Factory Defaults* on *page 41*. Each factory default is also shown after the syntax but before the example of each command description starting on *page 61*.

- The letter H in the Binary Byte or Binary Offset columns of the commands and logs tables represents the header length for that command or log, see *Section 1.1.3, Binary* on *page 25*.

- The number following 0x is a hexadecimal number.

- Default values shown in command tables indicate the assumed values when optional parameters have been omitted. Default values do not imply the factory default settings, see *"Factory Defaults" on page 41* for a list of factory default settings.

- Parameters surrounded by [and] are optional in a command or are required for only some instances of the command depending on the values of other parameters.

- Text displayed between < and > indicates the entry of a keystroke in the case of the command or an automatic entry in the case of carriage return <CR> and line feed <LF> in data output.

- In tables where no values are given they are assumed to be reserved for future use.

- Status words in ASCII logs are output as hexadecimal numbers and must be converted to binary format (and in some cases then also to decimal) to parse the fields because they are not fixed in 4-bits boundary. For an example of this type of conversion, please see the RANGE log, *Table 126* on *page 604*.

- Conversions and their binary or decimal results are always read from right to left. For a complete list of hexadecimal, binary and decimal equivalents, refer to the Unit Conversion information available on our website at www.novatel.com/support/search/.

- ASCII log examples may be split over several lines for readability. In reality, only a single [CR][LF] pair is transmitted at the end of an ASCII log.

You can download the most up-to-date version of this manual along with any addendums from the section of the NovAtel website.

Prerequisites

As this reference manual is focused on the OEM6 family commands and logging protocol, it is necessary to ensure the receiver has been properly installed and powered up according to the instructions outlined in the companion OEM6 Family Installation and Operation User Manual (OM-20000128) for OEM6 cards and the FlexPak6 enclosures and/or the ProPak6 User Manual (OM-20000148) before proceeding.

Chapter 1   Messages

1.1   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 and NMEA format messaging. For more information about message logs, refer to the contents of Chapters 2 and 3.

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 25)*.

Table 1 below, 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

Type	Binary Size (bytes)	Description
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
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
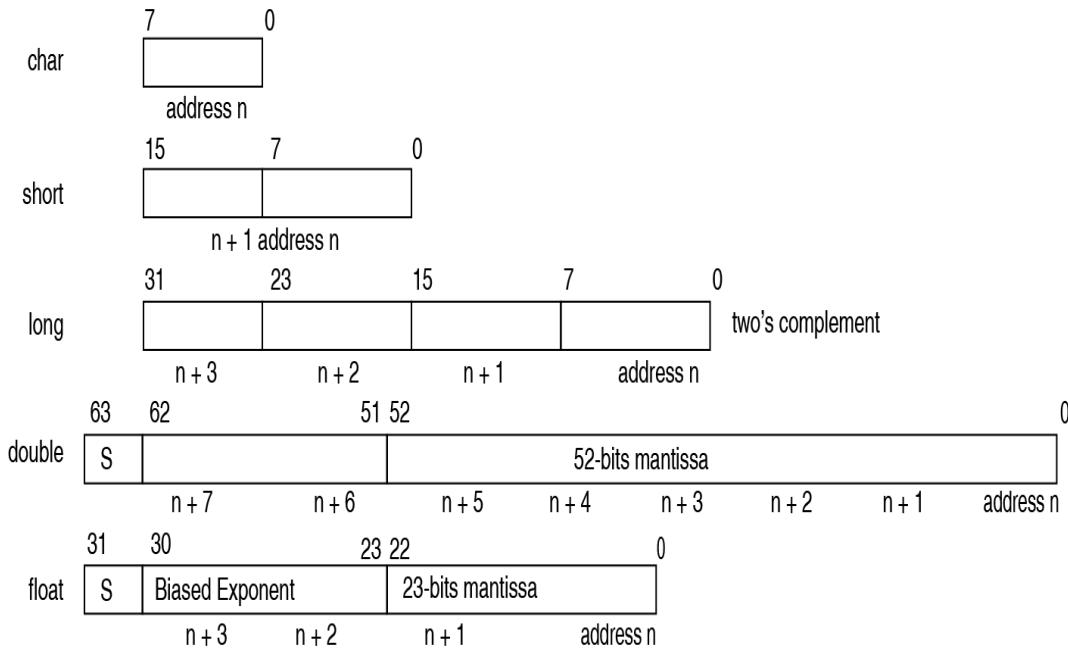
Figure 1: Byte Arrangements

Figure 1 shows the arrangement of bytes, within each field type, when used by IBM 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 *Table 1*. 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. See *Table 126, Channel Tracking Status* on *page 604* for a more detailed example.

1.1.1   *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 eight CRC digits.
    See *1.7, 32-Bit CRC* on *page 37* 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 4, Responses* on *page 754* 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* on *page 24*.

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 (lists are in the tables in *Command Reference* on *page 41* and *Log Reference* on *page 360*)	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, calculated once per second	Y
6	Time Status	Enum	The value indicates the quality of the GPS reference time (see *Table 8, GPS Reference Time Status* on *page 35*)	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 (see *Table 139, Receiver Status* on *page 661*)	Y
10	Reserved	Ulong	Reserved for internal use.	Y
11	Receivers 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,8
b0550a189aa6ff925386228f97eabf9c8047e34a70ec5a10e486e794a7a,8b0550a18a2effc2f
80061c2fffc267cd09f1d5034d3537affa28b6ff0eb*7a22f279
```

1.1.2   *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 entries are offset from the left hand side and start with '<'.

1.1.3   *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 *1.7, 32-Bit CRC* on *page 37* for the CRC algorithm) performed on all data including the header.

4. The header is in the format shown in *Table 3, Binary Message Header Structure* on *page 26.*

Table 3: 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 Lgth	Uchar	Length of the header	1	3	N
5	Message ID	Ushort	This is the Message ID number of the log (see the log descriptions in *Table 81, OEM6 Logs in Numerical Order* on *page 375* for the Message ID values of individual logs)	2	4	N

6	Message Type	Char	Bits 0-4 = Measurement source[a]
			Bits 5-6 = Format
			    00 = Binary
			    01 = ASCII
			    10 = Abbreviated ASCII, NMEA
			    11 = Reserved
			Bit 7 = Response bit
			(see *Section 1.2, page 32*)
			    0 = Original Message
			    1 = Response Message
				1	6	N

7	Port Address	Uchar	See *Table 4, Detailed Port Identifier* on *page 27* (decimal values >=32 may be used) (lower 8-bits only) [b]	1	7	N [c]
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, calculated once per second. 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 8, GPS Reference Time Status* on *page 35*).	1 [d]	13	N [e]
12	Week	Ushort	GPS reference week number	2	14	N [d]
13	ms	GPSec	Milliseconds from the beginning of the GPS reference week	4	16	N [d]

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 (see *Table 139, Receiver Status* on *page 661*)	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. Bits 0-4 are used to indicate the measurement source. For the OEM617D and FlexPak6D receivers, if bit 0 is set, that means the log is from the secondary antenna.

b. 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.

c. Recommended value is THISPORT (binary 192).

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

e. These time fields 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 4: 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
FILE_ALL	7	7	All virtual ports for logging to file[b]
ALL_PORTS	8	8	All virtual ports for all ports
XCOM1_ALL	9	9	All virtual ports for XCOM1
XCOM2_ALL	a	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[c]
XCOM3_ALL	11	17	All virtual XCOM3
COM4_ALL	13	19	All virtual ports for COM4 [b]
ETH1_ALL	14	20	All virtual ports for ETH1
IMU_ALL	15	21	All virtual ports for IMU
. . .			

ASCII Port Name	Hex Port Value	Decimal Port Value [a]	Description
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
WCOM1_ALL	1e	30	All virtual ports for WCOM1
. . .			
COM1	20	32	COM1, virtual port 0
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
FILE	e0	224	Virtual port 0 for logging to file [b]
. . .			
FILE_31	ff	255	Virtual port 31 for logging to file [b]
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
. . .			

ASCII Port Name	Hex Port Value	Decimal Port Value [a]	Description
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 [c]
. . .			
AUX_31	8bf	2239	AUX port, virtual port 31 [c]
XCOM3	9a0	2464	XCOM3, virtual port 0
. . .			
XCOM3_31	9bf	2495	XCOM3, virtual port 31
COM4	ba0	2976	COM4, virtual port 0 [b]
. . .			
COM4_31	bbf	3007	COM4, virtual port 31 [b]
ICOM1	fa0	4000	ICOM1, virtual port 0
PORT_ADDR_ETH1	ca0	3232	ETH1, virtual port 0
. . .			
PORT_ADDR_ETH1_31	cbf	3263	ETH1, virtual port 31
. . .			
PORT_ADDR_IMU	da0	3488	IMU, virtual port 0
. . .			
PORT_ADDR_IMU_31	dbf	3519	IMU, virtual port 31
. . .			
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
. . .			

ASCII Port Name	Hex Port Value	Decimal Port Value [a]	Description
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
. . .			
PORT_ADDR_WCOM1	16a0	5792	WCOM1, virtual port 0
. . .			
NCOM3_31	14bf	5311	NCOM3, virtual port 31
. . .			
PORT_ADDR_WCOM1_31	16bf	5823	WCOM1, virtual port 31
. . .			
COM5_ALL	16c0	5824	All virtual ports for COM5 [b]
COM6_ALL	16c1	5825	All virtual ports for COM6 [b]
BT1_ALL	16c2	5826	All virtual ports for the Bluetooth device [b]
COM5	17a0	6048	COM5, virtual port 0 [b]
. . .			
COM5_31	17bf	6079	COM5, virtual port 31 [b]
COM6	18a0	6304	COM6, virtual port 0 [b]
. . .			
COM6_31	18bf	6335	COM6, virtual port 31 [b]
BT1	19a0	6560	Bluetooth device, virtual port 0 [b]
. . .			
BT1_31	19bf	6591	Bluetooth device, virtual port 31 [b]
COM7_ALL	1a40	6720	All virtual ports for COM7
COM8_ALL	1a41	6721	All virtual ports for COM8
COM9_ALL	1a42	6722	All virtual ports for COM9
COM10_ALL	1a43	6723	All virtual ports for COM10
PORT_ADDR_COM7	1aa0	6816	COM7, virtual port 0
. . .			
PORT_ADDR_COM7_31	1abf	6847	COM7, virtual port 31

ASCII Port Name	Hex Port Value	Decimal Port Value [a]	Description
PORT_ADDR_COM8	1ba0	7072	COM8, virtual port 0
. . .			
PORT_ADDR_COM8_31	1bbf	7103	COM8, virtual port 31
PORT_ADDR_COM9	1ca0	7328	COM9, virtual port 0
. . .			
PORT_ADDR_COM9_31	1cbf	7359	COM1, virtual port 31
PORT_ADDR_COM10	1da0	7584	COM10, virtual port 0
. . .			
PORT_ADDR_COM10_31	1dbf	7615	COM10, virtual port 31

a. Decimal port values 0 through 16 are only available to the UNLOGALL command (refer to *page 342*) and cannot be used in the UNLOG command (refer to *page 340)* or in the binary message header (see *Table 3* on *page 26).*

b. These ports are only available on specific products.

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

> (i) 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.

Table 5 provides examples of where each port type might be used.

Table 5: Available Port Types

Port Type	Description	Example of where it might be used
AUX	Auxiliary "serial" ports	An additional UART serial port available only on certain platforms
BTx	Bluetooth ports	These ports are used to connect over Bluetooth devices, when the receiver is equipped with a BT device
COMx	Serial Port	UART serial ports. Used when there is a physical RS-232 or RS-422 connection to the receiver
ICOMx	Internet ports	These ports are used when establishing TCP or UDP connections to the receiver over a network
NCOMx	NTRIP ports	These ports are used when establishing NTRIP connections to the receiver over a network
USBx	USB "serial" ports	When the receiver is connected to an external host through USB, these ports are available
WCOMx	Web Server port	Ports used by Web Server applications, for receivers equipped with a web server
XCOMx	Virtual Port	A "port" not associated with a physical communications port. Normally used by API applications to request and retrieve logs internally

1.2   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 *page 184*). The response will be in the exact format you entered the message (that is, binary input = binary response).

1.2.1   Abbreviated Response

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

1.2.2   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

1.2.3 Binary Response

Similar to an ASCII response except that it follows the binary protocols, see *Table 6, Binary Message Response Structure* on *page 33*.

Table 7, Binary Message Sequence on *page 34* 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 6: Binary Message Response Structure

	Field #	Field Name	Field Type	Description	Binary Bytes	Binary Offset
B I N A R Y H E A D E R	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 4, Detailed Port Identifier* on *page 27*	1	7
	8	Message Length	Ushort	The length in bytes of the body of the message (not including the CRC)	2	8
	9	Sequence	Ushort	Normally 0	2	10
	10	Idle Time	Uchar	Idle time	1	12
	11	Time Status	Enum	*Table 8, GPS Reference Time Status* on *page 35*	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 139, Receiver Status* on *page 661*	4	20
	15	Reserved	Ushort	Reserved for internal use	2	24
	16	Receiver S/W Version	Ushort	Receiver software build number	2	26
I D	17	Response ID	Enum	The enumeration value corresponding to the message response (*Table 166, Response Messages* on *page 754*)	4	28
H E X	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 7: 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

1.3   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. Its actual GLONASS Slot Number is 6. The SATVIS log shows 43 (6+37). The actual GLONASS frequency is 1. The SATVIS log shows 8 (1+7). See also the SATVIS log on *page 670*.

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

1.4   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 8*).

Table 8: 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 *Section 1.5, Message Time Stamps* on *page 36* and the *SETTIMEBASE* command on page *307*.

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 a 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 startup and before any satellites are 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 (see *page 288*) or on receipt of an RTCAEPHEM message (see *page 634*) 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 (for more information, refer to the CLOCKADJUST command on *page 102*).

Once a position is known and range biases are being calculated, the internal clock model will begin modeling the range biases also known as 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 (see *page 102*). 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.

If a solution cannot be computed with the primary satellite system, it will attempt to use a backup system (if available). When the backup system is used and time is computed, the time status is set to FINEBACKUPSTEERING. If the position if lost and the range bias cannot be calculated, the time status is degraded to FREEWHEELING.

1.5   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. 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.

1.6 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.

1.7 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, 0x*42*, 0x*dc,* 0x*4c,* 0x*48*

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,1285,160578.000,00000020,5941,11
64;
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.
}
```

2.1 Command Formats

The receiver accepts commands in 3 formats as described in *Chapter 1*:

- Abbreviated ASCII
- ASCII
- Binary

Abbreviated ASCII is the easiest to use for your input. The other two formats include a CRC for error checking and are intended for use when interfacing with other electronic equipment.

The following are examples of the same command in each format:

Abbreviated ASCII Example:

```
LOG COM1 BESTPOSB ONTIME 1[CR]
```

ASCII Example:

```
LOGA,COM2,0,66.0,UNKNOWN,0,15.917,004c0000,5255,32858;COM1,
BESTPOSB,ONTIME,1.000000,0.000000,NOHOLD*F95592DD[CR]
```

Binary Example:

```
AA44121C 01000240 20000000 1D1D0000 29160000 00004C00 55525A80
20000000 2A000000 02000000 00000000 0000F03F 00000000 00000000
00000000 2304B3F1
```

2.2 Command Settings

There are several ways to determine the current command settings of the receiver:

1. Request an RXCONFIG log, see *page 657*. This log provides a listing of all commands issued to the receiver and their parameter settings. It also provides the most complete information.

2. For some specific commands, logs are available to indicate all their parameter settings. The LOGLIST log, see *page 526*, shows all active logs in the receiver beginning with the LOG command. The COMCONFIG log, see *page 112*, shows both the COM and INTERFACEMODE command parameter settings for all serial ports.

3. Request a log of the specific command of interest to show the parameters last entered for that command. The format of the log produced is exactly the same as the format of the specific command with updated header information.

> (i) Requesting a log for specific command is useful for most commands. For commands repeated with different parameters (for example, `SERIALCONFIG` and `LOG`), only the most recent set of parameters used is shown. To view all sets of parameters, try method 1 or 2 above.

Abbreviated ASCII Example:

```
log fix
<FIX COM1 0 45.0 FINE 1114 151898.288 00200000 dbfd 33123
<     NONE -10000.00000000000 -10000.00000000000 -10000.0000
```

2.3   Factory Defaults

When the receiver is first powered up or after a `FRESET` command (see *page 164*), all commands revert to their factory default settings. When you use a command without specifying its optional parameters, it may have a different command default than the factory default. The `SAVECONFIG` command (see *page 271*) can be used to save these defaults. Use the RXCONFIG log (see *page 657*) to reference any default command and log settings.

Ensure that all windows, other than the Console window, are closed in NovAtel's Connect user interface application before you issue the `SAVECONFIG` command.

> (i) FRESET STANDARD causes all previously stored user configurations saved to non-volatile memory to be erased (including Saved Config, Saved Almanac, Saved Ephemeris and L-Band-related data, excluding subscription information).

2.4   Command Reference

When a command is used without specifying its optional parameters, it may have a different command default than the factory default. Factory default settings for individual commands are stated in the following commands, organized alphabetically by command name.

Table 9: OEM6 Commands Sorted by Function

ASCII Name	Description	Message ID
Clock Information Status and Time		
ADJUST1PPS	Adjusts the receiver clock	429
CLOCKADJUST	Enables clock adjustments	15
CLOCKCALIBRATE	Adjusts clock steering parameters	430
CLOCKOFFSET	Adjusts for delay in 1PPS output	596

ASCII Name	Description	Message ID
EXTERNALCLOCK	Sets external clock parameters	230
SETAPPROXTIME	Sets an approximate GPS reference time	102
Communications, Control and Status		
APPLICATION	Starts/stops the application	413
COM	Sets COM port configuration	4
COMCONTROL	Controls the hardware control lines of the RS232 ports	431
DHCPCONFIG	Configure optional parameters for the DHCP Server	1710
DNSCONFIG	Configures DNS servers	1244
ECHO	Sets port echo	1247
ETHCONFIG	Configures Ethernet physical layer	1245
EVENTINCONTROL	Controls up to four Event-In input triggers	1637
EVENTOUTCONTROL	Controls Event-Out properties	1636
FREQUENCYOUT	Sets the output pulse train available on VARF	232
GENERATEDIFFCORRECTIONS	Sends differential corrections	1296
GENERATERTKCORRECTIONS	Sends RTK corrections	1260
ICOMCONFIG	Configures IP virtual COM port	1248
INTERFACEMODE	Sets interface type for a port, Receive (Rx)/Transmit (Tx)	3
IOCONFIG	Sets the behavior of multiplexed I/O pins	1663
IPCONFIG	Configures network IP settings	1243
IPSERVICE	Configures the availability of network ports and services	1575
LEDCONFIG	Configures LED indicators on the ProPak6	1498
LOG	Requests a log	1
LOGIN	Start a secure ICOM connection to the receiver	1671
LOGOUT	End a secure ICOM session started using the LOGIN command	1672
MARKCONTROL	Controls processing of the mark inputs	614
NTRIPCONFIG	Configures NTRIP	1249
NTRIPSOURCETABLE	Sets the NTRIPCASTER ENDPOINTS to be used for the SOURCETABLE log	1343
PPSCONTROL	Controls the PPS output	613
PPSCONTROL2	Controls the polarity, period, pulse width, and estimate error limit of the PPS output	1740
SEND	Sends ASCII message to a port	177

ASCII Name	Description	Message ID
SENDHEX	Sends non-printable characters to a port	178
SERIALCONFIG	Configures serial port settings	1246
SERIALPROTOCOL	Sets the protocol used by a serial port	1444
SETCANNAME	Sets the CAN name fields	1091
SOFTLOADCOMMIT	Completes the Softload process	475
SOFTLOADDATA	Sends firmware image data to the receiver for the SoftLoad process	1218
SOFTLOADFILE	Updates the receiver firmware using a *.hex or *.shex file that has been uploaded to the receiver	1302
SOFTLOADRESET	Initiates a new SoftLoad process	476
SOFTLOADSETUP	Sends configuration information to the receiver for the SoftLoad process	1219
SOFTLOADSREC	Sends an S-Record to the receiver for the SoftLoad process	477
TUNNELESCAPE	Breaks out of a tunnel	962
UNLOG	Removes a log from the logging control	36
UNLOGALL	Removes all logs from the logging control	38
Differential Rover/Base Station		
ALIGNAUTOMATION	Configures the ALIGN plug and play feature	1323
BASEANTENNAMODEL	Enters/changes the base antenna model	870
BASEANTENNAPCO	Sets the PCO model of the base receiver	1415
BASEANTENNAPCV	Sets the PCV model of the base receiver	1416
BASEANTENNATYPE	Sets the antenna type of the base receiver	1419
DGPSEPHEMDELAY	Sets DGPS ephemeris delay	142
DGPSTXID	Sets DGPS station ID	144
FIXPOSDATUM	Sets a datum position.	761
GENERATEALIGNCORRECTIONS	Configures the ALIGN Master and starts sending out ALIGN RTCA corrections through the specified port	1349
HPSEED	Specifies the initial position for OmniSTAR HP/XP	782
HPSTATICINIT	Sets the static initialization of OmniSTAR HP/XP	780
MOVINGBASESTATION	Enables a moving base station	763
POSAVE	Implements base station position averaging	173
POSTIMEOUT	Sets the position time out	612
PSRDIFFSOURCE	Sets the pseudorange correction source	493

ASCII Name	Description	Message ID
PSRDIFFSOURCETIMEOUT	Sets duration after differential corrections switched from one source to another	1449
PSRDIFFTIMEOUT	Sets maximum age of differential data	1450
RTKANTENNA	Specifies L1 phase center (PC) or ARP and enables/disables PC modeling	858
RTKCOMMAND	Resets the RTK filter or set the filter to default settings	97
RTKDYNAMICS	Sets up the RTK dynamics mode	183
RTKMATCHEDTIMEOUT	Sets the duration after matched RTK filter is reset	1447
RTKNETWORK	Specifies the RTK network mode	951
RTKQUALITYLEVEL	Chooses an RTK quality mode	844
RTKSOURCE	Sets the RTK correction source	494
RTKSOURCETIMEOUT	Sets duration after RTK corrections switched from one source to another	1445
RTKSVENTRIES	Sets the number of satellites in corrections	92
RTKTIMEOUT	Sets the maximum age of RTK data	910
SBASTIMEOUT	Sets the amount of time the receiver remains in an SBAS position if it stops receiving SBAS corrections	1001
SETBASERECEIVERTYPE	Specifies the base receiver type to aid GLONASS ambiguity fixing in RTK	1374
SETROVERID	Sets the Rover ID output in ROVERPOS, HEADING2, ALIGNBSLNXYZ and ALIGNBSLNENU logs	1135
SETRTCM16	Enters ASCII text for an RTCM data stream	131
SETRTCM36	Enters ASCII text with Russian characters	880
SETRTCMRXVERSION	Sets the expected RTCM message standard input	1216
SETRTCMTXVERSION	Sets the expected RTCM transmission standard input	1322
General Receiver Control		
ANTENNAPOWER	Controls power to Low Noise Amplifier (LNA) of an active antenna	98
AUTH	Adds the authorization code for a new model	49
COMVOUT	Controls power to the COM ports	779
DOSCMD	Issues DOS commands to the file system	1355
DUALANTENNAALIGN	Dual antenna ALIGN configuration	1761
DUALANTENNAPOWER	Controls the power of the Low Noise Amplifier (LNA) of an active antenna connected to the Ant 2 connector on a ProPak6 receiver	1639
FRESET	Resets the receiver to factory default	20
LOGFILE	Opens and closes log files in the internal flash memory	157

ASCII Name	Description	Message ID
MODEL	Switches the receiver to an existing authorized model	22
NVMRESTORE	Restores NVM data after an NVM failure	197
PROFILE	Configures multiple profiles in the non-volatile memory	1411
RESET	Resets the hardware	18
SAVECONFIG	Saves the current configuration	19
SAVEETHERNETDATA	Save the configuration data associated with an Ethernet interface	1679
SETADMINPASSWORD	Sets the administration password	1579
SETFILECOPYMODE	Configures the internal memory copy function	1581
SETPREFERREDNETIF	Set the network interface for DNS and default gateway configuration	1688
SETTIMEBASE	Sets the primary and backup systems for time base	1237
SETUTCLEAPSECONDS	Sets when future leap seconds take effect	1150
SOFTPOWER	Shuts down the receiver	213
STATUSCONFIG	Configures various status mask fields in RXSTATUSEVENT log	95
THISANTENNAPCO	Sets the PCO model of this receiver	1417
THISANTENNAPCV	Sets the PCV model of this receiver	1418
THISANTENNATYPE	Sets the antenna type of this receiver	1420
Position, Parameters and Solution Filtering Control		
BESTVELTYPE	Sets the velocity used in the BESTVEL and GPVTG logs	1678
DATUM	Chooses a DATUM name type	160
DIFFCODEBIASCONTROL	Enables or disables satellite differential code biases	913
FIX	Constrains receiver height or position	44
GGAQUALITY	Customizes the GPGGA GPS quality indicator	691
GLIDEINITIALIZATIONPERIOD	Configure the GLIDE initialization period	1424
HDTOUTTHRESHOLD	Controls the NMEA GPHDT log output	1062
HEADINGOFFSET	Adds specified offset to heading and pitch values	1082
IONOCONDITION	Sets ionospheric condition	1215
LOCALIZEDCORRECTIONDATUM	Sets a local datum	947
NMEATALKER	Sets the NMEA talker ID	861
NMEAVERSION	Sets the NMEA version for output	1574
OMNIUSEGLONASS	Enables/disables GLONASS in OmniSTAR	1199

ASCII Name	Description	Message ID
PDPFILTER	Enables, disables or resets the Pseudorange/Delta-Phase (PDP) filter	424
PDPMODE	Selects the PDP mode and dynamics	970
PDPVELOCITYOUT	Set the type of velocity used in the PDPVEL log	1324
PPPDYNAMICS	Sets the PPP dynamics mode	1551
PPPSEED	Control the seeding of the PPP filter	1544
PPPSOURCE	Specifies the PPP correction source	1707
PPPTIMEOUT	Sets the maximum age of the PPP corrections	1560
RAIMMODE	Configures RAIM mode	1285
RTKELEVMASK	Sets the RTK elevation mask	91
SBASCONTROL	Sets SBAS test mode and PRN	652
SETAPPROXPOS	Sets an approximate position	377
SETBESTPOSCRITERIA	Sets selection criteria for BESTPOS	839
SETDIFFCODEBIASES	Sets satellite differential code biases	687
SETIONOTYPE	Sets the ionospheric corrections model	711
SETTROPOMODEL	Sets the troposphere model used to correct ranges in the PSRPOS and PDPPOS solutions	1434
STEADYLINE	Configures position mode matching	1452
UNDULATION	Sets ellipsoid-geoid separation	214
USERDATUM	Sets user customized datum	78
USEREXPDATUM	Sets custom expanded datum	783
UALCONTROL	Setup User Accuracy levels	1627
UTMZONE	Sets UTM parameter	749
Satellite Tracking and Channel Control		
ASSIGN	Assigns individual satellite channel	27
ASSIGNALL	Assigns all satellite channels	28
ASSIGNLBAND	Sets L-Band satellite communication parameters	729
ASSIGNLBAND2	Sets L-Band satellite communication parameters	1200
ASSIGNLBANDBEAM	Configure L-Band Tracking	1733
BDSECUTOFF	Sets elevation cut-off angle for BeiDou satellites	1582
CNOUPDATE	Sets the C/No update rate and resolution	849
DATADECODESIGNAL	Enable/Disable navigation data decoding for GNSS signal	1532
DLLTIMECONST	Replaces the GLOCSMOOTH and CSMOOTH commands	1011

ASCII Name	Description	Message ID
DYNAMICS	Adjusts the tracking state transition time-out	258
ECUTOFF	Sets the GPS satellite tracking elevation cut-off angle	50
ELEVATIONCUTOFF	Sets the elevation cut-off angle for tracked satellites for a specified satellite system	1735
FORCEGLOL2CODE	Forces the receiver to track L2C or P-code	1217
FORCEGPSL2CODE	Forces the receiver to track L2C or P-code	796
GALECUTOFF	Sets the Galileo satellite elevation cut-off angle	1114
GLOECUTOFF	Sets the GLONASS satellite elevation cut-off angle	735
LOCKOUT	Prevents the receiver from using a satellite	137
LOCKOUTSYSTEM	Prevents the receiver from using a system	871
PPPCONVERGEDCRITERIA	Configures Decision for PPP Convergence	1566
QZSSECUTOFF	Sets the elevation cut-off angle for QZSS satellites	1350
SELECTCHANCONFIG	Sets the channel configuration	1149
SBASECUTOFF	Sets the SBAS satellite elevation cut-off	1000
TERRASTARPAYG	Enables use of TerraStar Pay-As-You-Go time	1749
TRACKSV	Overrides the automatic satellite/channel assignment for all satellites with manual instructions	1326
UNASSIGN	Unassigns a previously ASSIGNed channel	29
UNASSIGNALL	Unassigns all previously ASSIGNed channels	30
UNLOCKOUT	Reinstates a satellite in the solution	138
UNLOCKOUTALL	Reinstates all previously locked out satellites	139
UNLOCKOUTSYSTEM	Reinstates in the solution computation a system previously locked out	908
Waypoint Navigation		
MAGVAR	Sets the magnetic variation correction	180
SETNAV	Sets waypoints	162
Wireless Communication		
AIRPLANEMODE	Enables or disables Airplane mode	1674
BLUETOOTHCONFIG	Configures the Bluetooth parameters	1609
BLUETOOTHDISCOVERABILITY	Controls Bluetooth discoverability	1690
CELLULARCONFIG	Configure the cellular parameters	1683
WIFIAPCONFIG	Configure the Wi-Fi Access Point	1665
WIFICLICONFIG	Configure the Wi-Fi client	1614
WIFICLICONTROL	Controls the Wi-Fi client	1615

ASCII Name	Description	Message ID
WIFICONFIG	Configure the Wi-Fi radio power and operating mode	1617

Table 10: OEM6 Commands in Alphabetical Order

ASCII Name	Description	Message ID
ADJUST1PPS	Adjusts the receiver clock	429
AIRPLANEMODE	Enables or disables Airplane mode	1674
ALIGNAUTOMATION	Configures the ALIGN plug and play feature	1323
ANTENNAPOWER	Controls the power to Low Noise Amplifier (LNA) of an active antenna	98
APPLICATION	Starts/stops the application	413
ASSIGN	Assigns individual satellite channel	27
ASSIGNALL	Assigns all satellite channels	28
ASSIGNLBAND	Sets L-Band satellite communication parameters	729
ASSIGNLBAND2	Sets L-Band satellite communication parameters	1200
ASSIGNLBANDBEAM	Configure L-Band Tracking	1733
AUTH	Adds authorization code for new model	49
BASEANTENNAMODEL	Enters/changes base antenna model	870
BASEANTENNAPCO	Sets the PCO model of the base receiver	1415
BASEANTENNAPCV	Sets the PCV model of the base receiver	1416
BASEANTENNATYPE	Sets the antenna type of the base receiver	1419
BDSECUTOFF	Sets elevation cut-off angle for BeiDou satellites	1582
BESTVELTYPE	Sets the velocity used in the BESTVEL and GPVTG logs	1678
BLUETOOTHCONFIG	Configures the Bluetooth parameters	1609
BLUETOOTHDISCOVERABILITY	Controls Bluetooth discoverability	1690
CELLULARCONFIG	Configure the cellular parameters	1683
CLOCKADJUST	Enables clock adjustments	15
CLOCKCALIBRATE	Adjusts clock steering parameters	430
CLOCKOFFSET	Adjusts for delay in 1PPS output	596
CNOUPDATE	Sets C/No update rate and resolution	849
COM	Sets COM port configuration	4
COMCONTROL	Controls the hardware control lines of the RS232 ports	431
COMVOUT	Controls power to the COM ports	779
DATADECODESIGNAL	Enable/Disable navigation data decoding for GNSS signal	1532

ASCII Name	Description	Message ID
DATUM	Chooses a DATUM name type	160
DGPSEPHEMDELAY	Sets DGPS ephemeris delay	142
DGPSTXID	Sets DGPS station ID	144
DHCPCONFIG	Configure optional parameters for the DHCP Server	1710
DIFFCODEBIASCONTROL	Enables or disable satellite differential code biases	913
DLLTIMECONST	Replaces the GLOCSMOOTH and CSMOOTH commands	1011
DOSCMD	Issues DOS commands to the file system	1355
DNSCONFIG	Manually configures DNS servers	1244
DUALANTENNAALIGN	Dual antenna ALIGN configuration	1761
DUALANTENNAPOWER	Controls the power to the Low Noise Amplifier (LNA) of an active antenna connected to the Ant 2 connector on a ProPak6 receiver.	1639
DYNAMICS	Adjusts tracking state transition time-out	258
ECHO	Sets port echo	1247
ECUTEOFF	Sets GPS satellite tracking elevation cut-off angle	50
ELEVATIONCUTOFF	Sets the elevation cut-off angle for tracked satellites for a specified satellite system	1735
ETHCONFIG	Configures Ethernet physical layer	1245
EVENTINCONTROL	Controls up to four Event-In input triggers	1637
EVENTOUTCONTROL	Controls Event-Out properties	1636
EXTERNALCLOCK	Allows the OEM6 card to operate with an optional external oscillator	230
FIX	Constrains receiver height or position	44
FIXPOSDATUM	Sets the position in a specified datum	761
FORCEGLOL2CODE	Forces the receiver to track L2C or P-code	1217
FORCEGPSL2CODE	Forces the receiver to track L2C or P-code	796
FREQUENCYOUT	Sets the output pulse train available on VARF	232
FRESET	Resets the receiver to factory default	20
GALECUTOFF	Sets elevation cut-off angle for Galileo satellites	1114
GENERATEALIGNCORRECTIONS	Configures the ALIGN Master and starts sending out ALIGN RTCA corrections through the specified port	1349
GENERATEDIFFCORRECTIONS	Sends a preconfigured set of differential corrections	1296
GENERATERTKCORRECTIONS	Sends a preconfigured set of RTK corrections	1260
GGAQUALITY	Customizes the GPGGA GPS quality indicator	691

ASCII Name	Description	Message ID
GLIDEINITIALIZATIONPERIOD	Configure the GLIDE initialization period	1424
GLOECUTOFF	Sets the GLONASS satellite elevation cut-off angle	735
HDTOUTTHRESHOLD	Controls the NMEA GPHDT log output	1062
HEADINGOFFSET	Adds heading and pitch offset values	1082
HPSEED	Specifies the initial position for OmniSTAR HP/XP	782
HPSTATICINIT	Sets static initialization of OmniSTAR HP/XP	780
ICOMCONFIG	Configures IP virtual COM port	1248
INTERFACEMODE	Sets interface type for a port, Receive (Rx)/Transmit (Tx)	3
IOCONFIG	Sets the behavior of multiplexed I/O pins	1663
IONOCONDITION	Sets ionospheric condition	1215
IPCONFIG	Configures network IP settings	1243
IPSERVICE	Configures the availability of network ports and services	1575
LEDCONFIG	Configures LED indicators on the ProPak6	1498
LOCALIZEDCORRECTIONDATUM	Sets a local datum	947
LOCKOUT	Prevents the receiver from using a satellite by specifying its PRN	137
LOCKOUTSYSTEM	Prevents the receiver from using a system	871
LOG	Requests a log	1
LOGFILE	Opens and closes log files in the internal flash memory	157
LOGIN	Start a secure ICOM connection to the receiver	1671
LOGOUT	End a secure ICOM session started using the LOGIN command	1672
MAGVAR	Sets magnetic variation correction	180
MARKCONTROL	Controls processing of the mark inputs	614
MODEL	Switches receiver to a previously AUTHed model	22
MOVINGBASESTATION	Sets ability to use a moving base station	763
NMEATALKER	Sets the NMEA talker ID	861
NMEAVERSION	Sets the NMEA version for output	1574
NTRIPCONFIG	Configures NTRIP	1249
NTRIPSOURCETABLE	Sets the NTRIPCASTER ENDPOINTS used for the SOURCETABLE log	1343
NVMRESTORE	Restores NVM data after a failure in NVM	197
OMNIUSEGLONASS	Enables/disables GLONASS in OmniSTAR	1199

ASCII Name	Description	Message ID
PDPFILTER	Enables, disables or resets the Pseudorange/Delta-Phase (PDP) filter	424
PDPMODE	Selects the PDP mode and dynamics	970
PDPVELOCITYOUT	Set the type of velocity used in the PDPVEL log	1324
POSAVE	Implements base station position averaging	173
POSTIMEOUT	Sets the position time out	612
PPPCONVERGEDCRITERIA	Configures Decision for PPP Convergence	1566
PPPDYNAMICS	Sets the PPP dynamics mode	1551
PPPSEED	Control the seeding of the PPP filter	1544
PPPSOURCE	Specifies the PPP correction source	1707
PPPTIMEOUT	Sets the maximum age of the PPP corrections	1560
PPSCONTROL	Controls the PPS output	613
PPSCONTROL2	Controls the polarity, period, pulse width, and estimate error limit of the PPS output	1740
PROFILE	Configures multiple profiles in the non-volatile memory	1411
PSRDIFFSOURCE	Sets the pseudorange correction source	493
PSRDIFFSOURCETIMEOUT	Sets duration after differential corrections switched from one source to another	1449
PSRDIFFTIMEOUT	Sets maximum age of differential data	1450
QZSSECUTOFF	Sets the elevation cut-off angle for QZSS satellites	1350
RAIMMODE	Configures RAIM mode	1285
RESET	Performs a hardware reset	18
RTKANTENNA	Specifies L1 phase center (PC) or ARP and enable/disable PC modeling	858
RTKCOMMAND	Resets the RTK filter or sets the filter to default settings	97
RTKDYNAMICS	Sets up the RTK dynamics mode	183
RTKELEVMASK	Sets the RTK elevation mask	91
RTKMATCHEDTIMEOUT	Sets the duration after matched RTK filter is reset	1447
RTKNETWORK	Specifies the RTK network mode	951
RTKQUALITYLEVEL	Chooses an RTK quality mode	844
RTKSOURCE	Sets the RTK correction source	494
RTKSOURCETIMEOUT	Sets duration after RTK corrections switched from one source to another	1445
RTKSVENTRIES	Sets number of satellites in corrections	92

ASCII Name	Description	Message ID
RTKTIMEOUT	Sets maximum age of RTK data	910
SAVECONFIG	Saves current configuration	19
SAVEETHERNETDATA	Save the configuration data associated with an Ethernet interface	1679
SBASCONTROL	Sets SBAS test mode and PRN	652
SBASECUTOFF	Sets SBAS satellite elevation cut-off	1000
SBASTIMEOUT	Sets the amount of time the receiver remains in an SBAS position if it stops receiving SBAS corrections	1001
SELECTCHANCONFIG	Sets the channel configuration	1149
SEND	Sends ASCII message to a port	177
SENDHEX	Sends non-printable characters to a port	178
SERIALCONFIG	Configures serial port settings	1246
SERIALPROTOCOL	Sets the protocol used by a serial port	1444
SETADMINPASSWORD	Sets the administration password	1579
SETAPPROXPOS	Sets an approximate position	377
SETAPPROXTIME	Sets an approximate GPS reference time	102
SETBASERECEIVERTYPE	Specifies the base receiver type to aid GLONASS ambiguity fixing in RTK	1374
SETBESTPOSCRITERIA	Selects criteria for BESTPOS	839
SETCANNAME	Sets the CAN name fields	1091
SETDIFFCODEBIASES	Sets satellite differential code biases	687
SETFILECOPYMODE	Configures the internal memory copy function	1581
SETIONOTYPE	Sets the ionospheric corrections model	711
SETNAV	Sets waypoints	162
SETPREFERREDNETIF	Set the network interface for DNS and default gateway configuration	1688
SETROVERID	Sets the Rover ID output in ROVERPOS, HEADING2, ALIGNBSLNXYZ and ALIGNBSLNENU logs	1135
SETRTCM16	Enters ASCII text for RTCM data stream	131
SETRTCM36	Enters ASCII text with Russian characters	880
SETRTCMRXVERSION	Sets the expected RTCM standard input	1216
SETRTCMTXVERSION	Sets the RTCM transmission standard	1322
SETTIMEBASE	Sets primary and backup systems for time base	1237
SETTROPOMODEL	Sets the troposphere model used to correct ranges n the PSRPOS and PDPPOS solutions	1434

ASCII Name	Description	Message ID
SETUTCLEAPSECONDS	Sets when future leap seconds take effect	1150
SOFTLOADCOMMIT	Completes the Softload process	475
SOFTLOADDATA	Sends firmware image data to the receiver for the SoftLoad process	1218
SOFTLOADFILE	Updates the receiver firmware using a *.hex or *.shex file that has been uploaded to the receiver	1302
SOFTLOADRESET	Initiates a new SoftLoad process	476
SOFTLOADSETUP	Sends configuration information to the receiver for the SoftLoad process	1219
SOFTLOADSREC	Sends an S-Record to the receiver for the SoftLoad process	477
SOFTPOWER	Shuts down the receiver	213
STATUSCONFIG	Configures various status mask fields in RXSTATUSEVENT log	95
STEADYLINE	Configures position mode matching	1452
TERRASTARPAYG	Enables use of TerraStar Pay-As-You-Go time	1749
THISANTENNAPCO	Sets the PCO model of this receiver	1417
THISANTENNAPCV	Sets the PCV model of this receiver	1418
THISANTENNATYPE	Sets the antenna type of this receiver	1420
TRACKSV	Overrides the automatic satellite/channel assignment for all satellites with manual instructions	1326
TUNNELESCAPE	Breaks out of an established tunnel	962
UALCONTROL	Setup User Accuracy levels	1627
UNASSIGN	Unassigns a previously ASSIGNed channel	29
UNASSIGNALL	Unassigns all previously ASSIGNed channels	30
UNDULATION	Sets ellipsoid-geoid separation	214
UNLOCKOUT	Reinstates a satellite in the solution	138
UNLOCKOUTALL	Reinstates all previously locked out satellites	139
UNLOCKOUTSYSTEM	Reinstates in the solution computation a system previously locked out	908
UNLOG	Removes a log from the logging control	36
UNLOGALL	Removes all logs from the logging control	38
USERDATUM	Sets user customized datum	78
USEREXPDATUM	Sets custom expanded datum	783
UTMZONE	Sets UTM parameters	749
WIFIAPCONFIG	Configure the Wi-Fi Access Point	1665

ASCII Name	Description	Message ID
WIFICLICONFIG	Configure Wi-Fi client	1614
WIFICLICONTROL	Controls the Wi-Fi client	1615
WIFICONFIG	Configure the Wi-Fi radio power and operating mode	1617

Table 11: OEM6 Commands in Numerical Order

Message ID	ASCII Name	Description
1	LOG	Requests a log
3	INTERFACEMODE	Sets interface type for a port, Receive (Rx)/Transmit (Tx)
4	COM	Sets COM port configuration
15	CLOCKADJUST	Enables clock adjustments
18	RESET	Performs a hardware reset
19	SAVECONFIG	Saves current configuration
20	FRESET	Resets receiver to factory default
22	MODEL	Switches receiver to a previously AUTHed model
27	ASSIGN	Assigns individual satellite channel
28	ASSIGNALL	Assigns all satellite channels
29	UNASSIGN	Unassigns a previously ASSIGNed channel
30	UNASSIGNALL	Unassigns all previously ASSIGNed channels
36	UNLOG	Removes a log from the logging control
38	UNLOGALL	Removes all logs from the logging control
44	FIX	Constrains receiver height or position
49	AUTH	Adds authorization code for new model
50	ECUTOFF	Sets GPS satellite tracking elevation cut-off angle
78	USERDATUM	Sets user customized datum
91	RTKELEVMASK	Sets the RTK elevation mask
92	RTKSVENTRIES	Sets number of satellites in corrections
95	STATUSCONFIG	Configures various status mask fields in RXSTATUSEVENT log
97	RTKCOMMAND	Resets the RTK filter or sets the filter to default settings
98	ANTENNAPOWER	Controls power to Low Noise Amplifier (LNA) of an active antenna
102	SETAPPROXTIME	Sets an approximate GPS reference time
131	SETRTCM16	Enters ASCII text for RTCM data stream

Message ID	ASCII Name	Description
137	LOCKOUT	Prevents the receiver from using a satellite by specifying its PRN
138	UNLOCKOUT	Reinstates a satellite in the solution
139	UNLOCKOUTALL	Reinstates all previously locked out satellites
142	DGPSEPHEMDELAY	Sets DGPS ephemeris delay
144	DGPSTXID	Sets DGPS station ID
157	LOGFILE	Opens and closes log files in the internal flash memory
160	DATUM	Chooses a DATUM name type
162	SETNAV	Sets waypoints
173	POSAVE	Implements base station position averaging
177	SEND	Sends ASCII message to a port
178	SENDHEX	Sends non-printable characters to a port
180	MAGVAR	Sets magnetic variation correction
183	RTKDYNAMICS	Sets up the RTK dynamics mode
197	NVMRESTORE	Restores NVM data after a failure in NVM
213	SOFTPOWER	Shuts down the receiver
214	UNDULATION	Sets ellipsoid-geoid separation
230	EXTERNALCLOCK	Allows the OEM6 card to operate with an optional external oscillator
232	FREQUENCYOUT	Sets the output pulse train available on VARF
258	DYNAMICS	Adjusts tracking state transition time-out
377	SETAPPROXPOS	Sets an approximate position
413	APPLICATION	Starts/stops the application
424	PDPFILTER	Enables, disables or resets the Pseudorange/Delta-Phase (PDP) filter
429	ADJUST1PPS	Adjusts the receiver clock
430	CLOCKCALIBRATE	Adjusts clock steering parameters
431	COMCONTROL	Controls the hardware control lines of the RS232 ports
475	SOFTLOADCOMMIT	Completes the SoftLoad process
476	SOFTLOADRESET	Initiates a new SoftLoad process
477	SOFTLOADSREC	Sends an S-Record to the receiver for the SoftLoad process
493	PSRDIFFSOURCE	Sets the pseudorange correction source
494	RTKSOURCE	Sets the RTK correction source

Message ID	ASCII Name	Description
596	CLOCKOFFSET	Adjusts for delay in 1PPS output
612	POSTIMEOUT	Sets the position time out
613	PPSCONTROL	Controls the PPS output
614	MARKCONTROL	Controls processing of the mark inputs
652	SBASCONTROL	Sets SBAS test mode and PRN
687	SETDIFFCODEBIASES	Sets satellite differential code biases
691	GGAQUALITY	Customizes the GPGGA GPS quality indicator
711	SETIONOTYPE	Sets the ionospheric corrections model
729	ASSIGNLBAND	Sets L-Band satellite communication parameters
735	GLOECUTOFF	Sets the GLONASS satellite elevation cut-off angle
749	UTMZONE	Sets UTM parameters
761	FIXPOSDATUM	Sets the position in a specified datum
763	MOVINGBASESTATION	Sets ability to use a moving base station
779	COMVOUT	Controls power to the COM ports
780	HPSTATICINIT	Sets static initialization of OmniSTAR HP/XP
782	HPSEED	Specifies the initial position for OmniSTAR HP/XP
783	USEREXPDATUM	Sets custom expanded datum
796	FORCEGPSL2CODE	Forces the receiver to track L2C or P-code
839	SETBESTPOSCRITERIA	Selects criteria for BESTPOS
844	RTKQUALITYLEVEL	Chooses an RTK quality mode
849	CNOUPDATE	Sets C/No update rate and resolution
858	RTKANTENNA	Specifies L1 phase center (PC) or ARP and enable/disable PC modeling
861	NMEATALKER	Sets the NMEA talker ID
870	BASEANTENNAMODEL	Enters/changes base antenna model
871	LOCKOUTSYSTEM	Prevents the receiver from using a system
880	SETRTCM36	Enters ASCII text with Russian characters
908	UNLOCKOUTSYSTEM	Reinstates in the solution computation a system previously locked out
910	RTKTIMEOUT	Sets maximum age of RTK data
913	DIFFCODEBIASCONTROL	Enables or disable satellite differential code biases
947	LOCALIZEDCORRECTION-DATUM	Sets a local datum
951	RTKNETWORK	Specifies the RTK network mode
962	TUNNELESCAPE	Breaks out of an established tunnel

Message ID	ASCII Name	Description
970	PDPMODE	Selects the PDP mode and dynamics
1000	SBASECUTOFF	Sets SBAS satellite elevation cut-off
1001	SBASTIMEOUT	Sets the amount of time the receiver remains in an SBAS position if it stops receiving SBAS corrections
1011	DLLTIMECONST	Replaces the GLOCSMOOTH and CSMOOTH commands
1062	HDTOUTTHRESHOLD	Controls the NMEA GPHDT log output
1082	HEADINGOFFSET	Adds specified offset to heading and pitch values
1091	SETCANNAME	Sets the CAN name fields
1114	GALECUTOFF	Sets elevation cut-off angle for Galileo satellites
1135	SETROVERID	Sets the Rover ID output in ROVERPOS, HEADING2, ALIGNBSLNXYZ and ALIGNBSLNENU logs
1149	SELECTCHANGCONFIG	Sets the channel configuration
1150	SETUTCLEAPSECONDS	Sets when future leap seconds take effect
1199	OMNIUSEGLONASS	Enables/disables GLONASS in OmniSTAR
1200	ASSIGNLBAND2	Sets L-Band satellite communication parameters
1215	IONOCONDITION	Sets ionospheric condition
1216	SETRTCMRXVERSION	Sets the RTCM transmission standard
1217	FORCEGLOL2CODE	Forces the receiver to track L2C or P-code
1218	SOFTLOADDATA	Sends firmware image data to the receiver for the SoftLoad process
1219	SOFTLOADSETUP	Sends configuration information to the receiver for the SoftLoad process
1237	SETTIMEBASE	Sets primary and backup systems for time base
1243	IPCONFIG	Configures network IP settings
1244	DNSCONFIG	Manually configures DNS servers
1245	ETHCONFIG	Configures Ethernet physical layer
1246	SERIALCONFIG	Configures serial port settings
1247	ECHO	Sets port echo
1248	ICOMCONFIG	Configures IP virtual COM port
1249	NTRIPCONFIG	Configures NTRIP
1260	GENERATERTKCORRECTIONS	Sends a preconfigured set of RTK corrections
1285	RAIMMODE	Configures RAIM mode
1296	GENERATEDIFFCORRECTIONS	Sends a preconfigured set of differential corrections

Message ID	ASCII Name	Description
1302	SOFTLOADFILE	Updates the receiver firmware using a *.hex or *.shex file that has been uploaded to the receiver
1322	SETRTCMTXVERSION	Sets the expected RTCM transmission standard input
1323	ALIGNAUTOMATION	Configures the ALIGN plug and play feature
1324	PDPVELOCITYOUT	Set the type of velocity used in the PDPVEL log
1326	TRACKSV	Overrides the automatic satellite/channel assignment for all satellites with manual instructions
1343	NTRIPSOURCETABLE	Sets the NTRIPCASTER ENDPOINTS to be used for the SOURCETABLE log
1349	GENERATEALIGNCORRECTIONS	Configures the ALIGN Master and start sending out ALIGN RTCA corrections through the specified port
1350	QZSSECUTOFF	Sets the elevation cut-off angle for QZSS satellites
1355	DOSCMD	Issues DOS commands to the file system
1374	SETBASERECEIVERTYPE	Specifies the base receiver type to aid GLONASS ambiguity fixing in RTK
1411	PROFILE	Configures multiple profiles in the non-volatile memory
1415	BASEANTENNAPCO	Sets the PCO model of the base receiver
1416	BASEANTENNAPCV	Sets the PCV model of the base receiver
1417	THISANTENNAPCO	Sets the PCO model of this receiver
1418	THISANTENNAPCV	Sets the PCV model of this receiver
1419	BASEANTENNATYPE	Sets the antenna type of the base receiver
1420	THISANTENNATYPE	Sets the antenna type of this receiver
1424	GLIDEINITIALIZATIONPERIOD	Configure the GLIDE initialization period
1434	SETTROPOMODEL	Sets the troposphere model used to correct ranges in the PSRPOS and PDPPOS solutions
1444	SERIALPROTOCOL	Sets the protocol used by a serial port
1445	RTKSOURCETIMEOUT	Sets duration after RTK corrections switched from one source to another
1447	RTKMATCHEDTIMEOUT	Sets duration after which matched RTK filter is reset
1449	PSRDIFFSOURCETIMEOUT	Sets duration after differential corrections switched from one source to another
1450	PSRDIFFTIMEOUT	Sets maximum age of differential data
1452	STEADYLINE	Configures position mode matching
1498	LEDCONFIG	Configures LED indicators on the ProPak6
1532	DATADECODESIGNAL	Enable/Disable navigation data decoding for GNSS signal

Message ID	ASCII Name	Description
1544	PPPSEED	Control the seeding of the PPP filter
1551	PPPDYNAMICS	Sets the PPP dynamics mode
1560	PPPTIMEOUT	Sets the maximum age of the PPP corrections
1566	PPPCONVERGEDCRITERIA	Configures Decision for PPP Convergence
1574	NMEAVERSION	Sets the NMEA version for output
1575	IPSERVICE	Configures the availability of network ports and services
1579	SETADMINPASSWORD	Sets the administration password
1581	SETFILECOPYMODE	Configures the internal memory copy function
1582	BDSECUTOFF	Sets elevation cut-off angle for BeiDou satellites
1609	BLUETOOTHCONFIG	Configures the Bluetooth parameters
1614	WIFICLICONFIG	Configure Wi-Fi client
1615	WIFICLICONTROL	Controls the Wi-Fi client
1617	WIFICONFIG	Configure the Wi-Fi radio power and operating mode
1627	UALCONTROL	Setup User Accuracy levels
1636	EVENTOUTCONTROL	Controls Event-Out properties
1637	EVENTINCONTROL	Controls up to four Event-In input triggers
1639	DUALANTENNAPOWER	Controls the power to the Low Noise Amplifier (LNA) of an active antenna connected to the Ant 2 connector on a ProPak6 receiver.
1663	IOCONFIG	Sets the behavior of multiplexed I/O pins
1665	WIFIAPCONFIG	Configure the Wi-Fi Access Point
1671	LOGIN	Start a secure ICOM connection to the receiver
1672	LOGOUT	End a secure ICOM session started using the LOGIN command
1674	AIRPLANEMODE	Enables or disables Airplane mode
1678	BESTVELTYPE	Sets the velocity used in the BESTVEL and GPVTG logs
1679	SAVEETHERNETDATA	Save the configuration data associated with an Ethernet interface
1683	CELLULARCONFIG	Configure the cellular parameters
1688	SETPREFERREDNETIF	Set the network interface for DNS and default gateway configuration
1690	BLUETOOTHDISCOVERABILITY	Controls Bluetooth discoverability
1707	PPPSOURCE	Specifies the PPP correction source
1710	DHCPCONFIG	Configure optional parameters for the DHCP Server

Message ID	ASCII Name	Description
1733	ASSIGNLBANDBEAM	Configure L-Band Tracking
1735	ELEVATIONCUTOFF	Sets the elevation cut-off angle for tracked satellites for a specified satellite system
1740	PPSCONTROL2	Controls the polarity, period, pulse width, and estimate error limit of the PPS output
1749	TERRASTARPAYG	Enables use of TerraStar Pay-As-You-Go time
1761	DUALANTENNAALIGN	Dual antenna ALIGN configuration

2.4.1 ADJUST1PPS
Adjusts the receiver clock

OEM Platform: 615, 617, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use this command to adjust the receiver clock or as part of the procedure to transfer time between receivers. The number of pulses per second (PPS) is always set to 1 Hz with this command. It is typically used when the receiver is not adjusting its own clock and is using an external reference frequency.

To disable the automatic clock adjustment, refer to the `CLOCKADJUST` command on *page 102*. To configure the receiver to use an external reference oscillator, see the `EXTERNALCLOCK` command on *page 151*.

The ADJUST1PPS command can be used to:
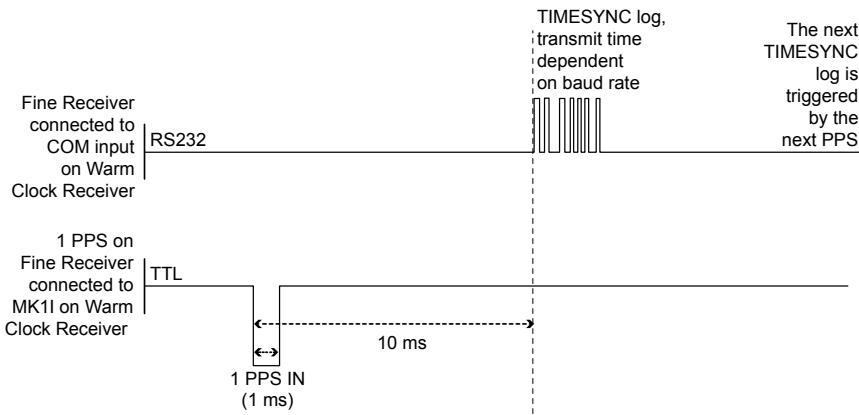
1. Manually shift the phase of the clock
2. Adjust the phase of the clock so the output 1PPS signal matches an external signal
3. Set the receiver clock close to that of another GNSS receiver
4. Set the receiver clock exactly in phase of another GNSS receiver

> ⓘ
> 1. The resolution of the clock synchronization is 50 ns.
> 2. To adjust the 1PPS output, when the receiver's internal clock is being used and the `CLOCKADJUST` command is enabled, use the `CLOCKOFFSET` command on *page 107*.
> 3. If the 1PPS rate is adjusted, the new rate does not start until the next second begins.

Figure 2 on *Page 61* shows the 1PPS alignment between a Fine and a Warm Clock receiver. See also the TIMESYNC log on *Figure 3.2.172* on *Page 734* and the *Transfer Time Between Receivers* section in the OEM6 Family Installation and Operation User Manual (OM-20000128).

Figure 2: 1PPS Alignment

The 1PPS is obtained from different receivers in different ways. For more information about 1PPS alignment, refer to the OEM6 Family Installation and Operation User Manual (OM-20000128) found on our website.

> (i) In *Figure 2, 1PPS Alignment on Page 61*, the examples are for the transfer of time. If you need position, you must be tracking satellites and your receiver must have a valid almanac.

Alternatively, the 1PPS signal can be set up to be output on the RTS signal of COM1, COM2 or COM3, or the DTR signal of COM2 using the COMCONTROL command, see *page 113*. The accuracy of the 1PPS is less using this method, but may be more convenient in some circumstances.

To find out the time of the last 1PPS output signal, use the TIMESYNCA/B output message, see *page 734*, which can be output serially on any available COM port, for example:

```
LOG COM1 TIMESYNCA ONTIME 1
```

Abbreviated ASCII Syntax: **Message ID: 429**

ADJUST1PPS mode [period] [offset]

Factory Default:

adjust1pps off

ASCII Example:

adjust1pps mark continuous 250

(i) Use the ADJUST1PPS command to synchronize two OEM6 cards in a primary/ secondary relationship to a common external clock.

At the Primary Receiver:

> log com2 timesynca ontime 1
>
> clockadjust disable
>
> externalclock ocxo 10 mhz (choose rubidium, cesium or user instead)
>
> externalclock frequency 10 (choose 5 instead if necessary)
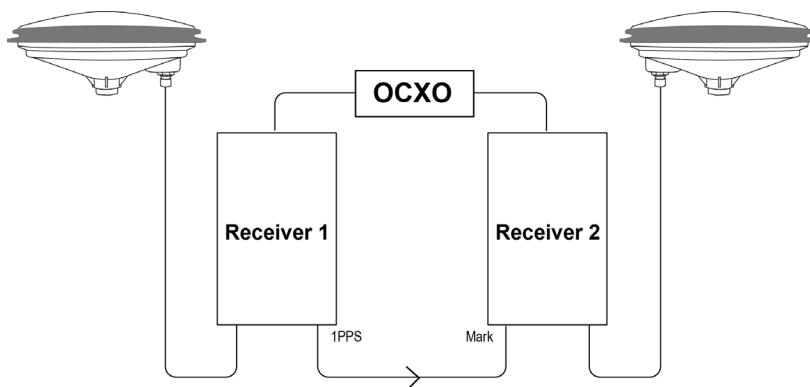
At the Secondary Receiver:

> interfacemode com2 novatel novatel
>
> clockadjust disable
>
> adjust1pps mark (or markwithtime or time depending on your connection (see Figure 3, ADJUST1PPS Connections on Page 64)
>
> externalclock ocxo 10 mhz (you can choose rubidium, cesium or user instead and choose 5 mHz instead if necessary)
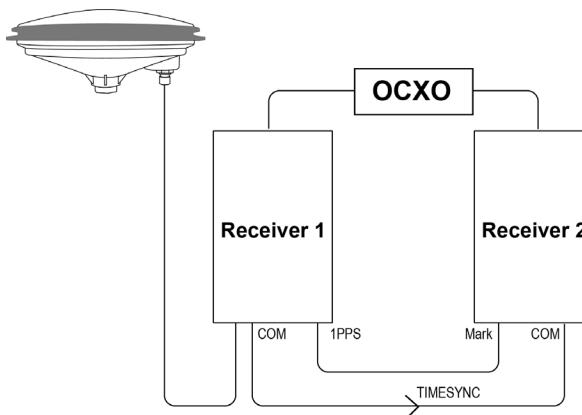
Connections:

Null modem cable connected from Primary COM2 to Secondary COM2

OCXO signal sent through a splitter to feed both the Primary and Secondary external clock inputs

Primary 1PPS connected to Secondary MKI
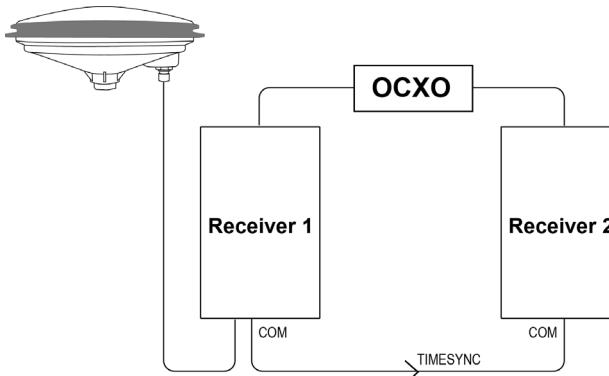
Connect everything before applying power. If power is applied and the OEM6 receivers have acquired satellites before the OCXO and/or 1PPS = MKI is set up, the times reported by the TIMESYNC logs still diverge. Note that after the clock model was stabilized at state 0, the time difference between the Primary and Secondary reported by the TIMESYNC log was less than 10 ns.

Figure 3: ADJUST1PPS Connections[1]

`adjust1pps mark`   (if Receiver 2 is not in coarsetime, the input is ignored)

`adjust1pps markwithtime`   (will get to finetime)

`adjust1pps time`   (will only get to coarsetime)

1. These examples are for the transfer of time. If you need position, you must be tracking satellites and your receiver must have a valid almanac.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	ADUST-1PPS header	-	-	This field contains the command name	-	H	0
2	mode	OFF	0	Disables ADJUST1PPS	Enum	4	H
		MANUAL	1	Immediately shifts the receivers time by the offset field in ns. The period field has no effect in this mode. This command does not affect the clock state			
		MARK[a]	2	Shifts the receiver time to align its 1PPS with the signal received in the MK1I port adjusted by the offset field in ns. The effective shift range is $\pm$ 0.5 s			
		MARKWITHTIME[b]	3	Shifts the receiver time to align its 1PPS with the signal received in the MK1I port adjusted by the offset field in ns, and sets the receiver TOW and week number, to that embedded in a received TIMESYNC log, see *page 734*. Also sets the receiver Time Status to that embedded in the TIMESYNC log, which must have arrived between 800 and 1000 ms prior to the MK1I event (presumably the 1PPS from the Primary), or it is rejected as an invalid message			
		TIME	4	If the receiver clock is not at least COARSEADJUSTED, this command enables the receiver to COARSE adjust its time upon receiving a valid TIMESYNC log in any of the ports. The clock state embedded in the TIMESYNC log must be at least FINE or FINESTEERING before it is considered. The receiver does not use the MK1I event in this mode			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	period	ONCE	0	The time is synchronized only once (default). The ADJUST1PPS command must be reissued if another synchronization is required	Enum	4	H+4
		CONTINUOUS	1	The time is continuously monitored and the receiver clock is corrected if an offset of more than 50 ns is detected			
4	offset	-2147483648 to +2147483647 (ns)		Allows the operator to shift the Secondary clock in 50 ns increments. In MANUAL mode, this command applies an immediate shift of this offset in ns to the receiver clock. In MARK and MARKWITHTIME mode, this offset shifts the receiver clock with respect to the time of arrival of the MK1I event. If this offset is zero, the Secondary aligns its 1PPS to that of the signal received in its MK1I port. For example, if this value was set to 50, then the Secondary would set its 1PPS 50 ns ahead of the input signal and if this value was set to -100 then the would set its clock to 100 ns behind the input signal. Typically, this offset is used to correct for cable delay of the 1PPS signal (default=0)	Long	4	H+8

a. Only the MK1I input can be used to synchronize the 1PPS signal. Synchronization cannot be done using the MK2I input offered on some receivers.

b. It is presumed that the TIMESYNC log, see *page 734*, was issued by a Primary GNSS receiver within 1000 ms but not less than 800 ms, of the last 1PPS event, see *Figure 2, 1PPS Alignment on Page 61* and *Section 3.2.172* on *page 734*. Also refer to the Transfer Time Between Receivers section in the OEM6 Family Installation and Operation User Manual (OM-20000128).

2.4.2 AIRPLANEMODE
Enable or disable airplane mode

OEM Platform: ProPak6

This command enables or disables Airplane mode. When the receiver is in Airplane mode, the Wi-Fi, cellular and Bluetooth radios are disabled.

> (i) Reception of GNSS signals is not affected by the state of Airplane mode.

The radios on the receiver keep track of two power states, their individual states and Airplane mode. When Airplane mode is ON, all radios (Wi-Fi, cellular and Bluetooth) are disabled regardless of their individual configurations. When Airplane mode is OFF, the radios follow their individual configurations.

> (i) When Airplane mode is enabled and then disabled, the radios are returned to their previously configured states. However, any connections that were made before Airplane mode was entered are not restored.

Abbreviated ASCII Syntax: **Message ID: 1674**

AIRPLANEMODE <enableenum>

Factory Default:

airplanemode disable

ASCII Example:

airplanemode enable

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	AIRPLANEMODE 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	EnableEnum	DISABLE	0	Disables Airplane mode. The Radio transmitters are turned on if the interface is configured. (Default)	Enum	4	H
		ENABLE	1	Enables Airplane mode. The Radio transmitters are turned off, regardless of their individual configurations.			

2.4.3 ALIGNAUTOMATION
Configures ALIGN plug-and-play feature

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command configures the ALIGN plug and play feature. Use this command to enable/disable the plug and play feature, to set the rover COM port to which master is connected, to set the baud rate for communication, to set the intended operation rate using this command and to enable/disable sending the HEADINGEXTB/ HEADINGEXT2B back to the Master receiver. Refer to the NovAtel application note APN-048 for details on HEADINGEXT (available on our website at www.novatel.com/ support/).

On issuing this command at ALIGN Rover, Rover will automatically sync with Master and configure it to send corrections at the specified baud rate and specified data rate.

> (i) This command should only be issued at ALIGN Rover.

Abbreviated ASCII Syntax: **Message ID: 1323**

ALIGNAUTOMATION option [comport] [baudrate][datarate][headingextboption]

Factory Default:

      alignautomation disable

Example:

      alignautomation enable com2 230400 10 on

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	ALIGNAUTOMATION 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	option	ENABLE	1	Enable or disable the plug-and-play feature	Enum	4	H
		DISABLE	0				
3	comport	COM1, COM2 or COM3		Rover COM port to which master is connected (default= COM2) (*Table 57, Page 284*)	Enum	4	H+4

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	baudrate	9600, 19200, 38400, 57600, 115200, 230400, 460800 or 921600		Intended baud rate for data transmission (default=230400)	Ulong	4	H+8
5	datarate	1, 2, 4, 5, 10 or 20 Hz		Rate at which heading output is required (default=10 Hz)	Ulong	4	H+12
6	headingextboption	ON or OFF		Enable or disable sending HEADINGEXTB/ HEADINGEXT2B back to the Master ((default ON)	Enum	4	H+16

2.4.4 ANTENNAPOWER
Controls power to the antenna

OEM Platform: 628, 638, FlexPak6, ProPak6

This command enables or disables the supply of electrical power from the internal power source of the receiver to the Low Noise Amplifier (LNA) of an active antenna. Refer to the OEM6 Family Installation and Operation User Manual (OM-20000128) for further information about supplying power to the antenna.

There are several bits in the receiver status that pertain to the antenna (see *Table 139, Receiver Status* on *page 661*). These bits indicate whether the antenna is powered (internally or externally) and whether it is open circuited or short circuited.

Abbreviated ASCII Syntax: **Message ID: 98**

ANTENNAPOWER switch

Factory Default:

antennapower on

ASCII Examples:

antennapower on
antennapower off
antennapower on3v3

> ⓘ For the OEM628 receiver, it is possible to supply power to the LNA of an active antenna either from the antenna port of the receiver itself or from an external source. The internal antenna power supply of the receiver can produce 5 VDC +/-5% at up to 100 mA. This meets the needs of any of NovAtel's dual-frequency GNSS antennas, so, in most cases, an additional LNA power supply is not required.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	ANTENNAPOWER 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	switch	OFF	0	Disables antenna power	Enum	4	H
		ON	1	Enables antenna power (5V)			
		ON3V3	2	Enables antenna power (3V)			

2.4.5   APPLICATION
Starts/stops the application

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to start, stop or remove the user application.

Abbreviated ASCII Syntax: **Message ID: 413**

APPLICATION command [parameter] [priority] [stacksize]

Factory Default:

> application start

ASCII Examples:

> application start
> application stop
> application remove

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Application 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	Command	STOP	0	Stops application	Enum	4	H
		START	1	Starts application			
		REMOVE	2	Removes application			
3	Parameter	0-		When starting the application, provide this value as a parameter to the application (default=0) 0=highest valid priority 21=lowest valid priority	Ulong	4	H+4
4	Priority	0-		When starting the application, this value sets the application priority (default=1)	Long	4	H+8
5	Stack size	0-		When starting the application, this value sets the application stack size of the initial task (default=1000)	Long	4	H+12

> ⓘ Refer to the OEM6 Family Application Programming Interface (API) User Guide (OM-20000140) for details about creating, loading and running the application. Go to www.novatel.com/support and contact a representative to acquire this manual.

2.4.6 ASSIGN
Assigns a channel to a PRN

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

> ⓘ
> 1. The `ASSIGN` command should only be used by advanced users.
>
> 2. Assigning SV channel sets the forced assignment bit in the channel tracking status field which is reported in the RANGE and TRACKSTAT logs.
>
> 3. Assigning a PRN to a SV channel does not remove the PRN from the search space of the automatic searcher; only the SV channel is removed (that is, the searcher may search and lock onto the same PRN on another channel). The automatic searcher only searches for PRNs 1 to 32 for GPS channels, PRNs 38 to 61 for GLONASS (where available), PRNs 1-36 for Galileo (where available), PRNs 120 to 138, 183-187 for SBAS channels and PRNs 1-30 for BeiDou.
>
> 4. GLONASS SVs cannot be assigned if there is no information on GLONASS frequencies and matching slot numbers.

This command may be used to aid in the initial acquisition of a satellite by allowing you to manually override the automatic satellite/channel assignment and reacquisition processes. The command specifies that the indicated tracking channel search for a specified satellite, at a specified Doppler frequency, within a specified Doppler window.

The instruction remains in effect for the specified SV channel and PRN, even if the assigned satellite subsequently sets. If the satellite Doppler offset of the assigned SV channel exceeds that specified by the window parameter of the `ASSIGN` command, the satellite may never be acquired or reacquired. If a PRN has been assigned to a channel and the channel is currently tracking that satellite, when the channel is set to AUTO tracking, the channel immediately idles and returns to automatic mode.

To cancel the effects of ASSIGN, issue one of the following:

- The `ASSIGN` command with the *state* set to *AUTO*
- The `UNASSIGN` command (starting on *page 333*)
- The `UNASSIGNALL` command (starting on *page 334*)

These immediately return SV channel control to the automatic search engine.

Table 12: Channel State

Binary	ASCII	Description
0	IDLE	Set the SV channel to not track any satellites
1	ACTIVE[a]	Set the SV channel active (default)
2	AUTO	Tell the receiver to automatically assign PRN codes to channels

a. A PRN number is required when using the ACTIVE channel state in this command.

Abbreviated ASCII Syntax: **Message ID: 27**

ASSIGN channel [state] [prn [doppler [doppler window]]]

ASCII Example 1:

        assign 0 active 29 0 2000

In example 1, the first SV channel is acquiring satellite PRN 29 in a range from -2000 Hz to 2000 Hz until the satellite signal is detected.

ASCII Example 2:

        assign 11 28 -250 0

SV channel 11 is acquiring satellite PRN 28 at an offset of -250 Hz only.

ASCII Example 3:

        assign 11 idle

SV channel 11 is idled and does not attempt to search for satellites.

> ⓘ   OEM6 cards have 4 channels available for SBAS. They automatically use the healthy GEO satellites with the highest elevations. Use the `ASSIGN` command to enter a GEO PRN manually.

> **617D** **FPk6D**   For the OEM617D and FlexPak6D receivers, when using the `ASSIGN` command for SV channels on the primary antenna, the SV channel count goes from 0 to N-1, where N is the number of channels in the primary antenna channel configuration. When using the `ASSIGN` command for channels on the secondary antenna, the SV channel count begins at N and goes to N+(M-1), where M is the number of channels in the secondary antenna SV channel configuration.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	ASSIGN header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see *1.1 Message Types* on *page 21*)	-	H	0
2	channel	0 to *n-1*, where *n* is the maximum number of channels in the current channel configuration		Desired SV channel number where channel 0 is the first SV channel. The last channel depends on your model configuration	Ulong	4	H
3	state	Refer to *Table 12*		Set the SV channel state. If a value is not given, the default of ACTIVE is used when the additional optional parameters are entered	Enum	4	H+4
4	prn	GPS: 1-32 SBAS: 120-138, 183-187 GLONASS: see *1.3 GLONASS Slot and Frequency Numbers* on *page 34*. Galileo: 1-36 QZSS: 193-197 BDS: 1-30		Optional satellite PRN code. A value must be entered if the state parameter is neither IDLE or AUTO	Ulong	4	H+8
5	Doppler	-100 000 to 100 000 Hz		Current Doppler offset of the satellite Note: Satellite motion, receiver antenna motion and receiver clock frequency error must be included in the calculation of Doppler frequency (default = 0)	Long	4	H+12
6	Doppler window	0 to 10 000 Hz		Error or uncertainty in the Doppler estimate above. Note: This is a ± value. Example: 500 for ± 500 Hz (default = 4500)	Ulong	4	H+16

2.4.7 ASSIGNALL
Assigns all channels to a PRN

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

ⓘ The ASSIGNALL command should only be used by advanced users.

This command is used to override the automatic satellite/channel assignment and reacquisition processes for all receiver channels with manual instructions.

Abbreviated ASCII Syntax: **Message ID: 28**

ASSIGNALL [system][state][prn [doppler [doppler window]]]

Table 13: Channel System

Binary	ASCII	Description
3	ALL	All systems
99	GPS	GPS system
100	SBAS	SBAS system
101	GLONASS	GLONASS system
102	GALILEO	GALILEO system
103	BeiDou	BeiDou system
104	QZSS	QZSS system

ⓘ Only GLONASS satellites that are in the almanac are available to assign using a slot number in the ASSIGN command. The possible range is still 38 to 61.

ASCII Example 1:

assignall glonass idle

In example 1, all GLONASS channels are idled out, essentially stopping the receiver from tracking GLONASS.

ASCII Example 2:

assignall glonass auto

In example 2, all GLONASS channels are enabled in auto mode. This enables the receiver to automatically assign channels to track the available GLONASS satellites.

ⓘ This command is the same as ASSIGN except that it affects all SV channels of the specified system.

.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	ASSIGN-ALL 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	system	See *Table 13, Channel System* on *page 75*		System that SV channel is tracking. If no value is specified, the value defaults to ALL	Enum	4	H
3	state	*Refer to Table 12, Channel State on page 73)*		Set the SV channel state	Enum	4	H+4
4	prn	GPS: 1-32 SBAS: 120-138, 183-187 GLONASS: see *1.3 GLONASS Slot and Frequency Numbers* on *page 34*. Galileo: 1-36 QZSS: 193-197 BDS: 1-30		Optional satellite PRN code. A value must be entered if the state parameter is neither IDLE or AUTO	Ulong	4	H+8
5	Doppler	-100 000 to 100 000 Hz		Current Doppler offset of the satellite Note: Satellite motion, receiver antenna motion and receiver clock frequency error must be included in the calculation of Doppler frequency.(default = 0)	Long	4	H+12
6	Doppler window	0 to 10 000 Hz		Error or uncertainty in the Doppler estimate above.This is a ± value (for example, 500 for ± 500 Hz) (default =4500)	Ulong	4	H+16

ⓘ These command examples are only applicable to specific receiver models.

If the system field is used with this command and the receiver has no channels configured with that channel system, the command is rejected.

2.4.8   ASSIGNLBAND
Sets L-Band satellite communication parameters

OEM Platform: 628, 638, FlexPak6, ProPak6

This command enables a receiver channel to track the specified OmniSTAR signal at a specified frequency and baud rate.

> ⓘ In addition to a NovAtel receiver with L-Band capability, a subscription to the OmniSTAR or use of other DGPS service is required. Contact NovAtel Inc. at www.novatel.com/support/ for details.

Abbreviated ASCII Syntax: **Message ID: 729**

ASSIGNLBAND mode [freq] [baud]

Factory Default:

> assignlband idle

ASCII Example:

> assignlband omnistar 1536782 1200

Beam Frequencies

The OmniSTAR beam frequency chart can be found at http://www.omnistar.com.

For example: Use OmniSTAR VBS

> assignlband omnistar <freq> 1200
> psrdiffsource omnistar

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	ASSIGNLBAND 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	mode	See *Table 14, Page 78*		Set the mode	Enum	4	H
3	freq[a]	1525000 to 1560000 or 1525000000 to 1560000000		L-Band service beam frequency of satellite (Hz or kHz). See also *Beam Frequencies* on *page 77* (default = 1545000)	Ulong	4	H+4
4	baud	300, 600, 1200 or 2400		Data rate for communication (bps) with L-Band satellite (default = 1200)	Ulong	4	H+8

a. The frequency assignment can be made in kHz or Hz. For example:
ASSIGNLBAND OMNISTAR 1557855000 1200
ASSIGNLBAND OMNISTAR 1557855 1200

Table 14: L-Band Mode

Binary	ASCII	Description
0	Reserved	
1	OMNISTAR	Track OmniSTAR satellites. A frequency baud rate must be specified
2	Reserved	
3	IDLE	Configure the receiver to track any L-Band satellites. The 'freq' and 'baud' fields are optional and do not need to be specified in this mode
4	OMNISTARAUTO[a]	Automatically select the best OmniSTAR beam to track based on the receiver's position. This requires the receiver to have a downloaded satellite list from an OmniSTAR satellite. Therefore, a manual assignment is necessary the first time is used on a new receiver. After collection, the satellite list is stored in NVM for subsequent auto assignments. Lists are considered valid for 6 months and are constantly updated while an OmniSTAR signal is tracking. If the receiver has a valid satellite list, it is reported in a status bit in the LBANDSTAT log, see *page 519*
5	OMNISTARNARROW	Track OmniSTAR satellite using a 1100 Hz search window on reacquisitions

a. The receiver will always track an available local beam over a global beam. The receiver constantly monitors the satellite list to ensure it is tracking the best one and automatically switches beams if it is not tracking the best one. You can view the satellite list by logging the OMNIVIS log, see *page 556*.

2.4.9   ASSIGNLBAND2
Sets L-Band satellite communication parameter

OEM Platform: 628, 638, FlexPak6, ProPak6

This command enables a receiver channel to track the specified OmniSTAR signal at a specified frequency and baud rate the same as the `ASSIGNLBAND` command but you can also specify the OmniSTAR satellite service ID number. In addition to a NovAtel receiver with L-Band capability, a subscription to the OmniSTAR or use of other DGPS service is required. Contact NovAtel for more details.

Abbreviated ASCII Syntax: **Message ID: 1200**

ASSIGNLBAND2 mode [freq] [baud] [id]

Factory Default:

assignlband2 idle

ASCII Example:

assignlband2 omnistar 1557855000 1200 c685

Beam Frequencies

Use OmniSTAR VBS

assignlband2 omnistar <freq> 1200
psrdiffsource omnistar

The OmniSTAR beam frequency chart can be found at http://www.omnistar.com.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	ASSIGNLBAND2 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	mode	See *Table 14, L-Band Mode* on *page 78*		Set the mode	Enum	4	H
3	freq[a]	1525000 to 1560000 or 1525000000 to 1560000000		L-Band service beam frequency of satellite (Hz or kHz). See also *Beam Frequencies* on *page 79* (default = 1545000)	Ulong	4	H+4
4	baud	300, 600, 1200 or 2400		Data rate for communication with L-Band satellite (bps) (default = 1200)	Ulong	4	H+8

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
5	ID	0x0, 0xC685, 0x2873 or 0xFCEE		OmniSTAR satellite service ID number (default=0x0)	Ulong	4	H+12

a. The frequency assignment can be made in kHz or Hz. For example:
    ASSIGNLBAND2 OMNISTAR 1557855000 1200 c685
    ASSIGNLBAND2 OMNISTAR 1557855 1200 c685

2.4.10   ASSIGNLBANDBEAM
Configure L-Band tracking

OEM Platform: 628, 638, FlexPak6, ProPak6

This command selects the beam that provides L-Band data and configures the L-Band tracking. It can be used to assign TerraStar, Veripos and OmniSTAR beams.

If the receiver has previously downloaded a beam table, then AUTO will select the beam from the satellite with the highest elevation.

The LBANDBEAMTABLE log (refer to *3.2.66, LBANDBEAMTABLE* on *page 516*) reports the TerraStar and Veripos beams known to the receiver. The OMNIVIS log (page *556*) reports the OmniSTAR beams known to the receiver.

Abbreviated ASCII Syntax: **Message ID: 1733**

ASSIGNLBANDBEAM [option] [name] [frequency] [baudrate] [dopplerwindow]

Factory Default:

assignlbandbeam idle

ASCII Examples:

ASSIGNLBANDBEAM AUTO

ASSIGNLBANDBEAM 98W

ASSIGNLBANDBEAM MANUAL 98w 1539902500 1200

Table 15: L-Band Assignment Option

ASCII	Binary	Description
IDLE	0	Idle the L-Band channel
AUTO	1	Allow receiver to auto select the beam based on almanac information
MANUAL	2	Manually assign a beam

Field #	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	ASSIGNLBAND BEAM header	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see Section 1.1, Message Types on Page 21)	-	H	0
2	Option	Assignment option (see Table 15 above) (manual=default)	Enum	4	H
3	Name	Beam name (empty string=default)	Char[8]	8	H+4
4	Frequency	Beam frequency in Hz or kHz (0=default)	Ulong	4	H+12
5	Baud rate	Data baud rate (0=default)	Ulong	4	H+16
6	Doppler window	Doppler window to search (6000=default)	Ulong	4	H+20

2.4.11 AUTH
Authorization code for different model

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to add or remove authorization codes from the receiver. Authorization codes are used to authorize models of software for a receiver. Models control the functionality the receiver provides. The receiver is capable of keeping track of 24 authorization codes at one time. The MODEL command (see *page 216*) can then be used to switch between authorized models. The VALIDMODELS log (see *page 738*) lists the current available models in the receiver. The AUTHCODES log (see *page 392*) lists all Authorization codes entered into the receiver. This simplifies the use of multiple software models on the same receiver.

If there is more than one valid model in the receiver, the receiver either uses the model of the last auth code entered via the AUTH command or the model that was selected by the MODEL command, whichever was done last. Adding an Authorization Code or using the MODEL command causes an automatic reset of the receiver. Removing an Authorization Code does not cause a reset.

> Removing an authorization code will cause the receiver to permanently lose this information.

Abbreviated ASCII Syntax: **Message ID: 49**

AUTH [state] part1 part2 part3 part4 part5 model [date]

Input Example:

auth add t48jf2,w25dbm,jh46bj,2wghmj,8jw5tw,g2sr0rccr,101114

> When you are ready to upgrade from one model to another, call 1-800-NOVATEL to speak with our Customer Support/Sales Personnel, who can provide the authorization code that unlocks the additional features of your GNSS receiver. This procedure can be performed at your work site and takes only a few minutes.
> Receiver models can also be downgraded. This is a two step handshaking process and is best performed in a location with e-mail access.

> Use the VERSION log to check the boot version. Boot version OEM060200RB0000 introduces support for signature authorization codes in addition to standard authorization codes. Standard authorization codes are firmware specific meaning new codes are required if the receiver firmware is updated. Signature authorization codes will work with any firmware version that has been digitally signed by NovAtel, removing the need for new codes after an upgrade. Digitally signed firmware files can be found on www.novatel.com/support. To update firmware on receivers with older boot versions, please contact NovAtel Customer Support.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	AUTH 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	state	REMOVE[a]	0	Remove the authcode from the system	Enum	4	H
		ADD	1	Add the authcode to the system (default)			
		ADD_DOWNLOAD[b]	4	Add the authcode to an image that has been downloaded via SoftLoad software but is not yet running			
3	part1	6 character ASCII string		Authorization code section 1	String [max. 16]	Variable [b]	H+4
4	part2	6 character ASCII string		Authorization code section 2	String [max. 16]	Variable [b]	H+20
5	part3	6 character ASCII string		Authorization code section 3	String [max. 16]	Variable [b]	H+36
6	part4	6 character ASCII string		Authorization code section 4	String [max. 16]	Variable [b]	H+52
7	part5	6 character ASCII string		Authorization code section 5	String [max. 16]	Variable [b]	H+68
8	model	Alpha numeric	Null terminated	Model name of the receiver	String [max. 16]	Variable [c]	H+84
9	date	Numeric	Null terminated	Expiry date entered as yymmdd in decimal	String [max 7]	Variable [b]	Variable

a. For this parameter, the Part1-Part5 fields can be entered as 0 0 0 0 0, and only the model name entered.
b. This option is valid only after SOFTLOADSTATUS indicates a SoftLoad is COMPLETE. Once the receiver has been reset and the new image is running the ADD option must be used to add a new authcode
c. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.12   BASEANTENNAMODEL
Enters/changes base antenna model

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

> (i) It is recommended that you use the BASEANTENNAPCO and BASEANTENNAPCV commands or the BASEANTENNATYPE command rather than the BASEANTENNAMODEL command.

This command allows you to enter or change an antenna model for a base receiver. Setting this value changes the appropriate field in RTCM23, RTCM1007 and RTCM1008 messages.You can set the antenna setupID to any value from 0-255.

Phase center offsets are entered as northing, easting and up. The Phase Center Variation (PCV) entries follow the NGS standard and correspond to the phase elevation at 5 degree increments starting at 90 degrees and decreasing to 0. All units are in mm.

> (i)   1. L1/L2 processing should include both L1 and L2 values or the resulting values might be incorrect. Since the phase measurement itself is corrected with the L1/L2 difference, failure to enter these values could result in bad position fixes.
>
> 2. It is recommended that you only enter antenna information if complete antenna model information is available. This information is best used in high precision static survey situations where antenna models are available for the base and rover receivers.
> To enter rover antenna information, use the THISANTENNAPCO and THISANTENNAPCV commands or the THISANTENNATYPE command.
> To enter the RTK antenna information, use the RTKANTENNA command.

Abbreviated ASCII Syntax: **Message ID: 870**

BASEANTENNAMODEL name sn setupid type [l1 offset n] [l1 offset e] [l1 offset up] [l1 var] [l2 offset n] [l2 offset e] [l2 offset up] [l2 var]

Factory Default:

baseantennamodel none none 0 none

ASCII Example:

baseantennamodel 702 nvh05410007 1 user

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	BASE-ANTENNA-MODEL 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	name			Antenna name	String[32]	Variable[a]	H
3	SN			Antenna serial number	String[32]	Variable[a]	Variable
4	setupID (0-255)			Setup identification - setting this value changes the appropriate field in RTCM23, RTCM1007 and RTCM1008, see pages *550, 515* and *515* respectively	Ulong	4	Variable
5	type[b]	NO ANTENNA	0	Antenna model type	Enum	4	Variable
		USER ANTENNA	1				
6	L1 offset N			L1 phase offsets northing (default = 0.0 mm)	Double	8	Variable
7	L1 offset E			L1 phase offsets easting (default = 0.0 mm)	Double	8	Variable
8	L1 offset UP			L1 phase offsets up (default = 0.0 mm)	Double	8	Variable
9	L1 var			L1 phase center variations (default = 0.0 *mm for all 19*)	Double [19]	152	Variable
10	L2 offset N			L2 phase offsets northing (default = 0.0 mm)	Double	8	Variable
11	L2 offset E			L2 phase offsets easting (default = 0.0 mm)	Double	8	Variable
12	L2 offset UP			L2 phase offsets up (default = 0.0 mm)	Double	8	Variable
13	L2 var			L2 phase center variations (default = 0.0 mm *for all 19*)	Double [19]	152	Variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

b. This should always be a user antenna when data is being entered manually for phase center offsets and/or phase center variation arrays.

2.4.13 BASEANTENNAPCO
Sets the PCO model of the base receiver

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use the `BASEANTENNAPCO` command to set the Phase Center Offsets (PCO) for a given frequency on the remote base receiver from which this receiver is receiving corrections. The Offsets are defined as North, East and Up from the Antenna Reference Point to the Frequency Phase Center in millimetres.

Abbreviated ASCII Syntax: **Message ID: 1415**

BASEANTENNAPCO frequency northoffset eastoffset upoffset[correctiontype][stationid]

ASCII Example:

baseantennapco gpsl1 0.61 1.99 65.64

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	BASEANTENNA PCO 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	Frequency	See *Table 16, Frequency Type* on *page 87*		The frequency that the phase center offsets are valid for.	Enum	4	H
3	North Offset			NGS standard Phase Center North Offset in millimetres.	Double	8	H+4
4	East Offset			NGS standard Phase Center East Offset in millimetres.	Double	8	H+12
5	Up Offset			NGS standard Phase Center Up Offset in millimetres.	Double	8	H+20
6	Correction Type	See *Table 50, DGPS Type* on *page 248*		Correction type (default = AUTO)	Enum	4	H+28
7	Station ID	Char [8] or ANY		ID string for the base station (default = ANY)	Char	8	H+32

Table 16: Frequency Type

Value	Name	Description
0	GPSL1	GPS L1 frequency
1	GPSL2	GPS L2 frequency
2	GLONASSL1	GLONASS L1 frequency
3	GLONASSL2	GLONASS L2 frequency
5	GPSL5	GPS L5 frequency
7	GALILEOE1	Galileo E1 frequency
8	GALILEOE5A	Galileo E5a frequency
9	GALILEOE5B	Galileo E5b frequency
10	GALILEOALTBOC	Galileo AltBOC frequency
11	BEIDOUB1	BeiDou B1 frequency
12	BEIDOUB2	BeiDou B2 frequency
13	QZSSL1	QZSS L1 frequency
14	QZSSL2	QZSS L2 frequency
15	QZSSL5	QZSS L5 frequency
16	QZSSLEX	QZSS LEX frequency
17	GALILEOE6	Galileo E6 frequency
18	BEIDOUB3	BeiDou B3 frequency

2.4.14 BASEANTENNAPCV
Sets the PCV model of the base receiver

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use the BASEANTENNAPCV command to set the Phase Center Variation (PCV) for a given frequency on the remote base receiver from which this receiver is receiving corrections. The Phase Center Variation entries follow the NGS standard and correspond to the phase elevation at 5 degree increments starting at 90 degrees and decreasing to 0.

Abbreviated ASCII Syntax: **Message ID: 1416**

BASEANTENNAPCV frequency [pcvarray] [correctiontype][stationid]

ASCII Example:

> baseantennapcv gpsl1 0.00 -0.020 -0.07 -0.15 -0.24 -0.34 -0.43 -0.51 -0.56 -0.61 -0.65 -0.69 -0.69 -0.62 -0.44 -0.13 0.28 0.70 1.02

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	BASEANTENNAPCV 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	Frequency	See *Table 16, Frequency Type* on *page 87*		The frequency for which the phase center variations are valid.	Enum	4	H
3	PCV Array			NGS standard 19 element array of phase center variations, in millimetres, in 5 degree elevation increments from 90 to 0. Defaults to zero for all elevation increments.	Double [19]	152	H+4
4	Correction Type	See *Table 50, DGPS Type* on *page 248*		Correction type defaults to AUTO	Enum	4	H+156
5	Base station ID	Char [8] or ANY		ID string defaults to ANY	Char	8	H+160

2.4.15 BASEANTENNATYPE
Sets the antenna type of the base receiver

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use the BASEANTENNATYPE command to set the antenna type of the remote base receiver from which this receiver is receiving corrections. The Antenna Type and Radome Type are the NGS names for the antenna.

> When the antenna type is set using this command, the receiver will look up and use the Phase Center Variations and Phase Center Offsets from an internal table.

Abbreviated ASCII Syntax: **Message ID: 1419**

BASEANTENNATYPE antenna type [radometype][correctiontype][stationid]

ASCII Example:

baseantennatype nov702

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	BASEANTENNA TYPE 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	Antenna Type	See *Table 17, Antenna Type* on *page 90*	NGS Antenna Name		Enum	4	H
3	Radome Type	See *Table 18, Radome Type* on *page 95*	NGS Radome Name (default=AUTO)		Enum	4	H+4
4	Correction Type	See *Table 50, DGPS Type* on *page 248*	Correction type (default=AUTO)		Enum	4	H+8
5	Base station ID	Char [8] or ANY	ID string (default=ANY)		Char	8	H+12

> The latest information can be obtained from the National Geodetic Survey (NGS) site www.ngs.noaa.gov/ANTCAL.

Table 17: Antenna Type

Value	Name	Description
0	NONE	No antenna model
1	USER	User antenna model
2	AUTO	Determine the antenna model from the RTK corrections (Not valid for THISANTENNATYPE)
3	AERAT2775_43	
4	AOAD_M_B	
5	AOAD_M_T	AOAD/M_T
6	AOAD_M_TA_NGS	AOAD/M_TA_NGS
7	APSAPS-3	
8	ASH700228A	
9	ASH700228B	
10	ASH700228C	
11	ASH700228D	
12	ASH700228E	
13	ASH700699.L1	
14	ASH700700.A	
15	ASH700700.B	
16	ASH700700.C	
17	ASH700718A	
18	ASH700718B	
19	ASH700829.2	
20	ASH700829.3	
21	ASH700829.A	
22	ASH700829.A1	
23	ASH700936A_M	
24	ASH700936B_M	
25	ASH700936C_M	
26	ASH700936D_M	
27	ASH700936E	
28	ASH700936E_C	
29	ASH700936F_C	

Value	Name	Description
30	ASH701008.01B	
31	ASH701073.1	
32	ASH701073.3	
33	ASH701933A_M	
34	ASH701933B_M	
35	ASH701933C_M	
36	ASH701941.1	
37	ASH701941.2	
38	ASH701941.A	
39	ASH701941.B	
40	ASH701945B_M	
41	ASH701945C_M	
42	ASH701945D_M	
43	ASH701945E_M	
44	ASH701945G_M	
45	ASH701946.2	
46	ASH701946.3	
47	ASH701975.01A	
48	ASH701975.01AGP	
49	JAV_GRANT-G3T	
50	JAV_RINGANT_G3T	
51	JAVRINGANT_DM	
52	JNSMARANT_GGD	
53	JPLD/M_R	
54	JPLD/M_RA_SOP	
55	JPSLEGANT_E	
56	JPSODYSSEY_I	
57	JPSREGANT_DD_E	
58	JPSREGANT_SD_E	
59	LEIAR10	
60	LEIAR25	
61	LEIAR25.R3	

Value	Name	Description
62	LEIAR25.R4	
63	LEIAS05	
64	LEIAX1202GG	
65	LEIAS10	
66	LEIAX1203+GNSS	
67	LEIAT202+GP	
68	LEIAT202-GP	
69	LEIAT302+GP	
70	LEIAT302-GP	
71	LEIAT303	
72	LEIAT502	
73	LEIAT503	
74	LEIAT504	
75	LEIAT504GG	
76	LEIATX1230	
77	LEIATX1230+GNSS	
78	LEIATX1230GG	
79	LEIAX1202	
80	LEIGG02PLUS	
81	LEIGS08	
82	LEIGS09	
83	LEIGS12	
84	3S-02-TSADM	
85	3S-02-TSATE	
86	LEIGS15	
87	LEIMNA950GG	
88	LEISR299_INT	
89	LEISR399_INT	
90	LEISR399_INTA	
91	MAC4647942	
92	MPL_WAAS_2224NW	
93	MPL_WAAS_2225NW	

Value	Name	Description
94	MPLL1_L2_SURV	
95	NAVAN2004T	
96	NAVAN2008T	
97	NAX3G+C	
98	NOV_WAAS_600	
99	NOV501	
100	NOV501+CR	
101	NOV502	
102	NOV502+CR	
103	NOV503+CR	
104	NOV531	
105	NOV531+CR	
106	NOV600	
107	NOV702	
108	NOV702GG	
109	NOV750.R4	
110	SEN67157596+CR	
111	SOK_RADIAN_IS	
112	SOK502	
113	SOK600	
114	SOK702	
115	SPP571212238+GP	
116	STXS9SA7224V3.0	
117	TOP700779A	
118	TOP72110	
119	TPSCR.G3	
120	TPSCR3_GGD	
121	TPSCR4	
122	TPSG3_A1	
123	TPSHIPER_GD	
124	TPSHIPER_GGD	
125	TPSHIPER_LITE	

Value	Name	Description
126	TPSHIPER_PLUS	
127	TPSLEGANT_G	
128	TPSLEGANT2	
129	TPSLEGANT3_UHF	
130	TPSODYSSEY_I	
131	TPSPG_A1	
132	TPSPG_A1+GP	
133	TRM14177.00	
134	TRM14532.00	
135	TRM14532.10	
136	TRM22020.00+GP	
137	TRM22020.00-GP	
138	TRM23903.00	
139	TRM27947.00+GP	
140	TRM27947.00-GP	
141	TRM29659.00	
142	TRM33429.00+GP	
143	TRM33429.00-GP	
144	TRM33429.20+GP	
145	TRM39105.00	
146	TRM41249.00	
147	TRM41249USCG	
148	TRM4800	
149	TRM55971.00	
150	TRM57970.00	
151	TRM57971.00	
152	TRM5800	
153	TRM59800.00	
154	TRM59800.80	
155	TRM59900.00	
156	TRMR8_GNSS	
157	TRMR8_GNSS3	

Table 18: Radome Type

Value	Name
0	NONE
1	SPKE
2	SNOW
3	SCIS
4	SCIT
5	OLGA
6	PFAN
7	JVDM
8	LEIT
9	LEIC
10	LEIS
11	MMAC
12	NOVS
13	TPSH
14	CONE
15	TPSD
16	TCWD
17	UNAV
18	TZGD

2.4.16   BDSECUTOFF
Sets elevation cut-off angle for BeiDou satellites

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the tracking elevation cut-off angle for BeiDou satellites.

> ⓘ
> 1. Use the ELEVATIONCUTOFF command (*page 144*) to set the cut-off angle for all other systems.
>
> 2. For the OEM617D and FlexPak6D receivers, this command applies to both the primary and secondary antennas.

Abbreviated ASCII Syntax: **Message ID: 1582**

BDSECUTOFF angle

Factory Default:

bdsecutoff 5.0

ASCII Example:

bdsecutoff 10.0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	BDSECUT OFF 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	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

2.4.17 BESTVELTYPE
Sets the velocity used in the BESTVEL and GPVTG logs

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command configures the source of the velocity that is output in the BESTVEL and GPVTG logs. Set the type to something other than BESTPOS when an unchanging velocity source with specific characteristics is needed.

The Doppler velocity is the highest-availability, lowest-latency velocity available from the receiver. Due to its low latency, it is also the noisiest velocity.

Abbreviated ASCII Syntax: **Message ID: 1678**

BESTVELTYPE mode

Factory Default:

        bestveltype bestpos

ASCII Example:

   BESTVELTYPE DOPPLER

Table 19: Velocity Types

ASCII	Binary	Description
BESTPOS	0	Use the velocity from the same positioning filter that is being used to fill BESTPOS and GPGGA
DOPPLER	1	Always fill BESTVEL using Doppler-derived velocities

Field #	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	BESTVELTYPE header	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see Section 1.1, Message Types on Page 21)	-	H	0
2	mode	Velocity type (see Table 19 above)	Enum	4	H

2.4.18   BLUETOOTHCONFIG
Configures Bluetooth® parameters

OEM Platform: ProPak6

Use the `BLUETOOTHCONFIG` command to configure Bluetooth operation parameters.

> ⓘ This command can be saved with the `SAVECONFIG` command.

Abbreviated ASCII Syntax: **Message ID: 1609**

BLUETOOTHCONFIG bluetoothconfiguration switch value

Factory Default:

off

ASCII Example:

bluetoothconfig power on

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	BLUETOOTHCONFIG header	Command header		H	0
2	bluetooth configuration parameter	Parameter option (see *Table 20*)	Enum	4	H
3	value1	Switch option (see *Table 20*)	String	12	H+4
4	value2	Reserved	String [100]	100	H+16

Table 20: Supported BLUETOOTHCONFIG Parameters

No	Bluetooth Configuration	Value1	Value2	Notes
1	power	ON	n/a	Powers on the Bluetooth radio
		OFF	n/a	Powers off the Bluetooth radio

2.4.19 BLUETOOTHDISCOVERABILITY
Controls Bluetooth discoverability

OEM Platform: ProPak6

This command enables or disables the discoverability option of the Bluetooth module. When Bluetooth discoverability is enabled, devices can discover and pair with the ProPak6. When Bluetooth discoverability is disabled, devices that have previously paired with the Propak6 can still pair but new pairings cannot be made.

When the Bluetooth module is powered on, discoverability is disabled by default.

ⓘ   This command can not be saved with the SAVECONFIG command.

ⓘ   When Bluetooth discoverability is enabled, throughput on both the Bluetooth interface (BT) and any active Wi-Fi interfaces (ICOM, FTP, etc) is reduced.

Abbreviated ASCII Syntax: **Message ID: 1690**

BLUETOOTHDISCOVERABILITY onoffenum [reserved]

Factory Default:

off

ASCII Example:

bluetoothdiscoverability on

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	BLUETOOTH DISCOVERABILITY 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	OnOffEnum	OFF	0	Disable discoverability	Enum	4	H
		ON	1	Enable discoverability			
3	RESERVED				Ulong	4	H + 4

2.4.20 CELLULARCONFIG
Configures cellular parameters

OEM Platform: ProPak6

Use the `CELLULARCONFIG` command to configure cellular parameters, such as APN.

> ℹ This command can be saved with the `SAVECONFIG` command.

Abbreviated ASCII Syntax: **Message ID: 1683**

CELLULARCONFIG cellularconfiguration [value1] [value2]

Factory Default:

        cellularconfig power off

ASCII Example:

        cellularconfig power on

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	CELLULAR CONFIG Header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively			0
2	Cellular Configuration	See *Table 21* on *page 101*		Parameter to configure	Enum	4	H
3	Value1	Alpha numeric	Null terminated	First parameter value The valid range of values depends on the parameter being configured (see *Table 21* on *page 101*)	String (Max 256 bytes)	Variable[a]	H + 4
4	Value2	Alpha numeric	Null terminated	Second parameter value The valid range of values depends on the parameter being configured (see *Table 21* on *page 101*)	String (Max 100 bytes)	Variable[a]	Variable

  a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 21: CELLULARCONFIG Parameters

Binary Value	ASCII Value	Value1	Value2	Notes
Cellular Configuration				
1	power	ON	n/a	Powers on the cellular radio and enables telephony.
		OFF	n/a	Disables telephony, powers off the cellular radio. (The default is OFF.)
2	apn	\<name\>	n/a	Sets the Access Point Name, e.g., internet.com. This string is limited to 99 characters plus 1 null byte to terminate the string.
3	username	\<username\>	n/a	Sets the APN username, e.g., wapuser. This string is limited to 255 characters plus 1 null byte to terminate the string.
4	password	\<password\>	n/a	Sets the APN password, e.g., wappassword. This string is limited to 255 characters plus 1 null byte to terminate the string.
5	data	ON	n/a	Enables data connectivity on the configured APN. (The default is ON.)
		OFF	n/a	Disables data connectivity on the configured APN.
6	nettype	GSM	n/a	Configures the radio to use GSM networks only.
		UMTS	n/a	Configures the radio to use UMTS networks only.
		ANY	n/a	Configures the radio to use any available network. (The default is ANY.)
7	dataroam	ON	n/a	Enables data connectivity on the configured APN when the radio is roaming.
		OFF	n/a	Disables data connectivity on the configured APN when the radio is roaming. (The default is OFF)

ASCII Examples:

Enable/disable cellular telephony:
> cellularconfig power on
> cellularconfig power off

Configure the APN. The factory default APN is blank.
> cellularconfig apn \<apn name\>
> cellularconfig user \<username\>
> cellularconfig password \<password\>

Enable/disable cellular data connectivity using configured APN:
> cellularconfig data on
> cellularconfig data off

Enable disable data connectivity while roaming outside of home network:
> cellularconfig dataroam on
> cellularconfig dataroam off

2.4.21 CLOCKADJUST
Enables clock adjustments

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6D

All oscillators have some inherent drift. By default, the receiver attempts to steer the receiver's clock to accurately match GPS reference time. Use the CLOCKADJUST command to disable this function. The TIME log can then be used to monitor clock drift.

ⓘ
1. The CLOCKADJUST command should only be used by advanced users.

2. If the CLOCKADJUST command is ENABLED and the receiver is configured to use an external reference frequency (set in the EXTERNALCLOCK command, see *page 151*, for an external clock - TCXO, OCXO, RUBIDIUM, CESIUM, or USER), then the clock steering process takes over the VARF output pins and may conflict with a previously entered FREQUENCYOUT command, see *page 161*.

3. When using the EXTERNALCLOCK and CLOCKADJUST commands together, issue the EXTERNALCLOCK command first to avoid losing satellites.

4. When disabled, the range measurement bias errors continue to accumulate with clock drift.

5. Pseudorange, carrier phase and Doppler measurements may jump if the CLOCKADJUST mode is altered while the receiver is tracking.

6. When disabled, the time reported on all logs may be offset from GPS reference time. The 1PPS output may also be offset. The amount of this offset may be determined from the TIME log, see *page 731*.

7. A discussion on GPS reference time may be found in *Section 1.4 GPS Reference Time Status* on *page 35.*

Abbreviated ASCII Syntax: **Message ID: 15**

CLOCKADJUST switch

Factory Default:

   clockadjust enable

ASCII Example:

   clockadjust disable

ⓘ The CLOCKADJUST command can be used to calibrate an internal oscillator. Disable the CLOCKADJUST mode in order find out what the actual drift is from the internal oscillator. Watch the CLOCKMODEL log to see the drift rate and adjust the oscillator until the drift stops.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	CLOCKADJUST 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	switch	DISABLE	0	Disallow adjustment of internal clock	Enum	4	H
		ENABLE	1	Allow adjustment of internal clock			

2.4.22 CLOCKCALIBRATE
Adjusts clock steering parameters

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to adjust the control parameters of the clock steering loop. The receiver must be enabled for clock steering before these values can take effect. Refer to the CLOCKADJUST command, see *page 102*, to enable or disable clock steering.

To disable the clock steering process, issue the CLOCKADJUST DISABLE command.

The current values used by the clock steering process are listed in the CLOCKSTEERING log, see *page 436*.

> The values entered using the CLOCKCALIBRATE command are saved to non-volatile memory (NVM). To restore the values to their defaults, the FRESET CLKCALIBRATION command must be used. Issuing FRESET without the CLKCALIBRATION parameter will not clear the values (see *page 164* for more details).

Abbreviated ASCII Syntax: **Message ID: 430**

CLOCKCALIBRATE [mode][period] [width] [slope] [bandwidth]

ASCII Example:

    clockcalibrate auto

> The receiver by default steers its INTERNAL VCTCXO but can be commanded to control an EXTERNAL reference oscillator. Use the EXTERNALCLOCK command, see *page 151*, to configure the receiver to use an external reference oscillator. If the receiver is configured for an external reference oscillator and configured to adjust its clock, then the clock steering loop attempts to steer the external reference oscillator through the use of the VARF signal. Note that the clock steering control process conflicts with the manual FREQUENCYOUT command, see *page 161*. It is expected that the VARF signal is used to provide a stable reference voltage by the use of a filtered charge pump type circuit (not supplied).

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	CLOCKCALIBRATE 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	mode	SET	0	Sets the period, pulsewidth, slope and bandwidth values into NVM for the currently selected steered oscillator (INTERNAL or EXTERNAL)	Enum	4	H
		AUTO	1	Forces the receiver to do a clock steering calibration to measure the slope (change in clock drift rate with a 1 bit change in pulse width) and required pulsewidth to zero the clock drift rate. After the calibration, these values along with the period and bandwidth are entered into NVM and are then used from this point forward on the selected oscillator			
		OFF	2	Terminates a calibration process currently underway (default)			
3	period	0 to 262144		Signal period in 25 ns steps. Frequency Output = 40,000,000 / Period (default = 4400)	Ulong	4	H+4
4	pulsewidth	The valid range for this parameter is 10% to 90% of the period		Sets the initial pulse width that should provide a near zero drift rate from the selected oscillator being steered. The valid range for this parameter is 10% to 90% of the period. The default value is 2200. If this value is not known, (in the case of a new external oscillator) then it should be set to ½ the period and the mode should be set to AUTO to force a calibration (default=2200)	Ulong	4	H+8

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
5	slope			This value should correspond to how much the clock drift changes with a 1 bit change in the pulsewidth m/s/bit. The default values for the slope used for the INTERNAL and EXTERNAL clocks is -2.0 and -0.01 respectively. If this value is not known, then its value should be set to 1.0 and the mode should be set to AUTO to force a calibration. Once the calibration process is complete and using a slope value of 1.0, the receiver should be recalibrated using the measured slope and pulsewidth values (see the CLOCKSTEERING log, on *page 436*). This process should be repeated until the measured slope value remains constant (less than a 5% change) (default=-2.0)	Float	4	H+12
6	bandwidth			This is the value used to control the smoothness of the clock steering process. Smaller values result in slower and smoother changes to the receiver clock. Larger values result in faster responses to changes in oscillator frequency and faster start up clock pull in. The default values are 0.03 and 0.001 Hz respectively for the INTERNAL and EXTERNAL clocks (default=0.03)	Float	4	H+16

2.4.23 CLOCKOFFSET
Adjusts for delay in 1PPS output

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to remove a delay in the PPS output. The PPS signal is delayed from the actual measurement time due to two major factors:

- A delay in the signal path from the antenna to the receiver
- An intrinsic delay through the RF and digital sections of the receiver

The second delay is automatically accounted for by the receiver using a nominal value determined for each receiver type. However, since the delay from the antenna to the receiver cannot be determined by the receiver, an adjustment cannot automatically be made. The CLOCKOFFSET command can be used to adjust for this delay.

Abbreviated ASCII Syntax: **Message ID: 596**

CLOCKOFFSET offset

Factory Default:

clockoffset 0

ASCII Example:

clockoffset -15

> (i) There may be small variances in the delays for each cable or card. The CLOCKOFFSET command can be used to characterize each setup. For example, for a cable with a delay of 10 ns, the offset can be set to -10 to remove the delay from the PPS output.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	CLOCKOFFSET header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see *Message Types* on *page 21*)	-	H	0
2	offset	±200		Specifies the offset in nanoseconds	Long	4	H

2.4.24   CNOUPDATE
Sets the C/No update rate

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the C/No update rate.

Abbreviated ASCII Syntax:　　　　　　　　**Message ID: 849**

CNOUPDATE rate

Factory Default:

cnoupdate default

ASCII Example (rover):

cnoupdate 20hz

> (i)   Use the CNOUPDATE command for higher resolution update rate of the C/No measurements of the incoming GNSS signals. By default, the C/No values are calculated at approximately 4 Hz but this command allows you to increase that rate to 20 Hz.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	CNO-UPDATE 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	rate	DEFAULT	0	C/No update rate: 0 = Turn off C/No enhancement default = 4 Hz 1 = 20 Hz C/No updates	Enum	4	H
		20HZ	1				

2.4.25 COM
COM port configuration control V123

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

> ▼ The SERIALCONFIG command supersede the COM command (refer to *page 282*). Also refer to the ECHO command on *page 140*. For backwards compatibility, the COM command is still supported. Only the SERIALCONFIG command will be updated to support new or expanded functionality. Customers are urged to migrate to the SERIALCONFIG command.

This command permits you to configure the receiver's asynchronous serial port communications drivers.

The current COM port configuration can be reset to its default state at any time by sending it two hardware break signals of 250 milliseconds each, spaced by fifteen hundred milliseconds (1.5 seconds) with a pause of at least 250 milliseconds following the second break. This will:

- Stop the logging of data on the current port (see UNLOGALL on *Page 342)*
- Clear the transmit and receive buffers on the current port
- Return the current port to its default settings (see *Page 41* for details)
- Set the interface mode to NovAtel for both input and output (see the INTERFACEMODE command on *Page 184*)

See also *Section 2.3, Factory Defaults* on *page 41* for a description of the factory defaults, and the COMCONFIG log on *Page 291*.

> ⓘ 1. The COMCONTROL command, see *Page 113*, may conflict with handshaking of the selected COM port. If handshaking is enabled, then unexpected results may occur.
>
> 2. Baud rates higher than 115,200 bps are not supported by standard PC hardware. Special PC hardware may be required for higher rates, including 230400 bps, 460800 bps and 921600 bps. Also, some PC's have trouble with baud rates beyond 57600 bps.

Abbreviated ASCII Syntax: **Message ID: 4**

com [port] bps [parity[databits[stopbits[handshake[echo[break]]]]]]

Factory Default:

> com com1 9600 n 8 1 n off on
> com com2 9600 n 8 1 n off on
> com com3 9600 n 8 1 n off on
> com aux 9600 n 8 1 n off on

ASCII Example:

 COM COM1,57600,N,8,1,N,OFF,ON

Watch for situations where the COM ports of two receivers are connected together and the baud rates do not match. Data transmitted through a port operating at a slower baud rate may be misinterpreted as break signals by the receiving port if it is operating at a higher baud rate. This is because data transmitted at the lower baud rate is stretched relative to the higher baud rate. In this case, configure the receiving port to have break detection disabled using the COM command.

> Use the COM command before using the INTERFACEMODE command on each port. Turn break detection off using the COM command to stop the port from resetting because it is interpreting incoming bits as a break command.

> Refer to the following tables under the SERIALCONFIG command:
>
> *Table 57, COM Port Identifiers* on *page 284*
> *Table 58, Parity* on *page 284*
> *Table 59, Handshaking* on *page 284*

> The OEMV-3 AUX port does not support hardware handshaking. Only transmit and receive lines exist for the AUX port on the OEMV-3.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	COM 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	See *Table , Abbreviated ASCII Syntax: Message ID: 1246* on *page 283*		Port to configure. (default = THISPORT)	Enum	4	H
3	bps/baud	300, 600, 900, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, or 230400		Communication baud rate (bps). Bauds of 460800 and 921600 are also available on COM1 of OEMV-2-based products.	ULong	4	H+4
4	parity	See *Table 58* on *page 284*		Parity	Enum	4	H+8
5	databits	7 or 8		Number of data bits (default = 8)	ULong	4	H+12
6	stopbits	1 or 2		Number of stop bits (default = 1)	ULong	4	H+16
7	handshake	See *Table 59* on *page 284*		Handshaking	Enum	4	H+20
8	echo	OFF	0	No echo (default)	Enum	4	H+24
		ON	1	Transmit any input characters as they are received			
9	break	OFF	0	Disable break detection	Enum	4	H+28
		ON	1	Enable break detection (default)			

2.4.26 COMCONFIG

> (i) This command is replaced with the SERIALCONFIG command. See page *282* for more information.

2.4.27   COMCONTROL
Controls the RS232 hardware control lines

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to control the hardware control lines of the RS232 ports. The TOGGLEPPS mode of this command is typically used to supply a timing signal to a host PC computer by using the RTS or DTR lines. The accuracy of controlling the COM control signals is better than 900 μs. The other modes are typically used to control custom peripheral devices.

> ⓘ   1.   If handshaking is disabled, any of these modes can be used without affecting regular RS232 communications through the selected COM port. However, if handshaking is enabled, it may conflict with handshaking of the selected COM port, causing unexpected results.
>
> 2.   The PULSEPPSLOW control type cannot be issued for a TX signal.
>
> 3.   Only PULSEPPSHIGH, FORCEHIGH and FORCELOW control types can be used for a TX signal.

Abbreviated ASCII Syntax: **Message ID: 431**

COMCONTROL port signal control

Factory Default:

> comcontrol com1 rts default
> comcontrol com2 rts default
> comcontrol com3 rts default

ASCII Example 1:

> serialconfig com1 9600 n 8 1 n (to disable handshaking)
> comcontrol com1 rts forcelow
> comcontrol com2 dtr togglepps

ASCII Example 2:

> comcontrol com1 rts togglepps
> comcontrol com2 rts togglepps
> comcontrol com3 rts togglepps

ASCII Example 3:

To set a break condition on AUX:

> comcontrol aux tx forcelow

A break condition remains in effect until it is cleared.

To clear a break condition on AUX:

       comcontrol com1 tx default

or

       comcontrol com1 tx forcehigh

Table 22: Tx, DTR and RTS Availability

	Tx Available On	**DTR Available On**	**RTS Available On**
OEM615	COM1 and COM2	N/A	N/A
OEM628	COM1 and COM2	N/A	COM1 and COM2
OEM638	COM1, COM2, COM3, COM6	N/A	COM1, COM2 and COM3

ⓘ COM1 on the OEM628 is user configurable for RS-422. Refer to the *Technical Specifications* appendix and also the *Connecting Data Communication Equipment* section of the [OEM6 Family Installation and Operation User Manual](#) (OM-20000128).

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	COM CONTROL 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	COM1	1	RS232 port to control. Valid ports are COM1, COM2 and COM3	Enum	4	H
		COM2	2				
		COM3	3				
		COM6	32	COM6 is OEM638 only			
3	signal	RTS	0	COM signal to control. The controllable COM signals are RTS, DTR and TX. See also *Table 22, Tx, DTR and RTS Availability* on *page 114*	Enum	4	H+4
		DTR	1				
		TX	2				

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	control	DEFAULT	0	Disables this command and returns the COM signal to its default state	Enum	4	H+8
		FORCEHIGH	1	Immediately forces the signal high			
		FORCELOW	2	Immediately forces the signal low			
		TOGGLE	3	Immediately toggles the current sate of the signal			
		TOGGLE PPS	4	Toggles the state of the selected signal within 900 μs after each 1PPS event. The state change of the signal lags the 1PPS by an average value of 450 μs. The delay of each pulse varies by a uniformly random amount less than 900 μs			
		PULSEPPS LOW	5	Pulses the line low at a 1PPS event and to high 1 ms after it. Not for TX			
		PULSEPPS HIGH	6	Pulses the line high for 1 ms at the time of a 1PPS event			

2.4.28   COMVOUT
Controls power on the COM ports

OEM Platform: ProPak6

Use this command to control power to the Propak6 COM1 and COM2 ports.

When COMVOUT is used to turn on power for a COM port, power from the ProPak6 power connector is applied to pin 4 of the COM port (COM1 or COM2).

> Power is provided to pin 4 at the same voltage as the power supply connected to the ProPak6 power connector.
>
> To prevent damage to the Propak6 or the far-end device connected to COM1 and/or COM2, ensure the connections are correct before issuing this command.

Abbreviated ASCII Syntax: **Message ID: 779**

COMVOUT port switch

ASCII Example:

comvout com1 on

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	COMVOUT 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	COM1	1	Selects the COM1 port	Enum	4	H
		COM2	2	Selects the COM2 port			
3	Switch	OFF	0	Disables power on pin 4 of the selected port (by default, COMVOUT is disabled on the ports)	Enum	4	H + 4
		ON	1	Enables power on pin 4 of the selected port			

2.4.29 DATADECODESIGNAL
Enable/Disable navigation data decoding for GNSS signal

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use this command to enable or disable the data decode of any GNSS signal. By default, data decode is enabled on all GNSS signals except for GPS and QZSS L2C and L5 signals. This command allows the user to select the signals to be decoded.

> (i) L2P(Y) data is not decoded.

Abbreviated ASCII Syntax: **Message ID: 1532**

DATADECODESIGNAL signaltype switch

Abbreviated ASCII Example:

datadecodesignal gpsl2c enable

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	DATADECODE SIGNAL 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	signal type	See *Table 25, Signal Type* on *page 130*		GNSS Signal Type	Enum	4	H
3	switch	Disable	0	Enable or disable the data decoding	Enum	4	H+4
		Enable	1				

2.4.30   DATUM
Chooses a datum name type

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to select the geodetic datum for operation of the receiver. If not set, the factory default value is WGS84. See the USERDATUM command for user definable datums. The datum you select causes all position solutions to be based on that datum.

The transformation for the WGS84 to Local used in the OEM6 family is the Bursa-Wolf transformation or reverse Helmert transformation. In the Helmert transformation, the rotation of a point is counter clockwise around the axes. In the Bursa-Wolf transformation, the rotation of a point is clockwise. Therefore, the reverse Helmert transformation is the same as the Bursa-Wolf.

See *Table 23, Page 119* for a complete listing of all available predefined datums. The offsets in the table are from the local datum to WGS84.

Abbreviated ASCII Syntax:	**Message ID: 160**

DATUM datum

Factory Default:

datum wgs84

ASCII Example:

datum csrs

Also, as an example, you can achieve spatial integrity with Government of Canada maps and surveys if the coordinates are output using the CSRS datum (Datum ID# 64).

Table 23, Page 119 contains the internal ellipsoid and transformation parameters used in the receiver. The values contained in these tables were derived from the following DMA reports:

1	TR 8350.2	Department of Defense World Geodetic System 1984 and Relationships with Local Geodetic Systems - Revised March 1, 1988
2	TR 8350.2B	Supplement to Department of Defense World Geodetic System 1984 Technical Report - Part II - Parameters, Formulas, and Graphics for the Practical Application of WGS84 - December 1, 1987
3	TR 8350.2	Department of Defense World Geodetic System 1984 National Imagery and Mapping Agency Technical Report, Third Addition, Amendment 1 - January 3, 2000

ⓘ By default, NovAtel receivers output positions in WGS84, with the following
exceptions:

EGNOS, TerraStar, Veripos and OmniSTAR use ITRF2008, which is coincident
with WGS84 at about the decimetre level.

Table 23: Reference Ellipsoid Constants

ELLIPSOID	ID CODE	a (metres)	1/f	f
Airy 1830	AW	6377563.396	299.3249646	0.00334085064038
Modified Airy	AM	6377340.189	299.3249646	0.00334085064038
Australian National	AN	6378160.0	298.25	0.00335289186924
Bessel 1841	BR	6377397.155	299.1528128	0.00334277318217
Clarke 1866	CC	6378206.4	294.9786982	0.00339007530409
Clarke 1880	CD	6378249.145	293.465	0.00340756137870
Everest (India 1830)	EA	6377276.345	300.8017	0.00332444929666
Everest (Brunei & E.Malaysia)	EB	6377298.556	300.8017	0.00332444929666
Everest (W.Malaysia & Singapore)	EE	6377304.063	300.8017	0.00332444929666
Geodetic Reference System 1980	RF	6378137.0	298.257222101	0.00335281068118
Helmert 1906	HE	6378200.0	298.30	0.00335232986926
Hough 1960	HO	6378270.0	297.00	0.00336700336700
International 1924	IN	6378388.0	297.00	0.00336700336700
Parameters of the Earth	PZ-90.02	6378136.0	298.26	0.00335280374302
South American 1969	SA	6378160.0	298.25	0.00335289186924
World Geodetic System 1972	WD	6378135.0	298.26	0.00335277945417
World Geodetic System 1984	WE	6378137.0	298.257223563	0.00335281066475

Table 24: 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

Table 24: Datum Transformation Parameters

Datum ID#[a]	NAME	DX[b]	DY[b]	DZ[b]	DATUM DESCRIPTION	ELLIPSOID
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
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

Table 24: Datum Transformation Parameters

Datum ID#[a]	NAME	DX[b]	DY[b]	DZ[b]	DATUM DESCRIPTION	ELLIPSOID
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
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

Table 24: Datum Transformation Parameters

Datum ID#[a]	NAME	DX[b]	DY[b]	DZ[b]	DATUM DESCRIPTION	ELLIPSOID
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 Kauaii[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 starting on *page 345*. The following logs report the datum used according to the OEM card Datum ID column: BESTPOS, BESTUTM, MATCHEDPOS and PSRPOS.
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.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	DATUM 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	Datum Type	See *Table 24, Datum Transformation Parameters* on *page 120*		The datum to use	Enum	4	H

2.4.31 DGPSEPHEMDELAY
Sets DGPS ephemeris delay

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the ephemeris delay when operating as a base station. The ephemeris delay sets a time value by which the base station continues to use the old ephemeris data. A delay of 120 to 300 seconds typically ensures that the rover stations have collected updated ephemeris. After the delay period is passed, the base station begins using new ephemeris data.

The factory default of 120 seconds matches the RTCM standard.

> ⓘ The RTCA Standard stipulates that a base station shall wait five minutes after receiving a new ephemeris before transmitting differential corrections to rover stations that are using the RTCA standard. This time interval ensures that the rover stations have received the new ephemeris and have computed differential positioning based upon the same ephemeris. Therefore, for RTCA base stations, the recommended ephemeris delay is 300 seconds.

Abbreviated ASCII Syntax: **Message ID: 142**

DGPSEPHEMDELAY delay

Factory Default:

> dgpsephemdelay 120

ASCII Example (base):

> dgpsephemdelay 120

> ⓘ When using differential corrections, the rover receiver must use the same set of broadcast ephemeris parameters as the base station generating the corrections. The Issue of Ephemeris Data (IODE) parameter is transmitted as part of the differential correction so that the rover can guarantee that its and the base station ephemerides match. The DGPSEPHEMDELAY parameter should be large enough to ensure that the base station is not using a new set of ephemerides that has not yet been received at the rover receiver.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	DGPSEPHEMDELAY 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	delay	0 to 600 s		Minimum time delay before new ephemeris is used	Ulong	4	H

2.4.32 DGPSTXID
Sets DGPS station ID

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the station ID value for the receiver when it is transmitting corrections. This allows for the easy identification of which base station was the source of the data.

For example, if you want to compare RTCM and RTCMV3 corrections, you would be easily able to identify their base stations by first setting their respective DGPSTXID values.

Abbreviated ASCII Syntax: **Message ID: 144**

DGPSTXID type id

Factory Default:

dgpstxid auto any

ASCII Examples:

dgpstxid rtcm 2	- using an rtcm type and id
dgpstxid cmr 30	- using a cmr type and id
dgpstxid cmr any	- using the default cmr id
dgpstxid rtca d36d	- using an rtca type and id
dgpstxid rtcmv3 2050	- using an rtcmv3 type and id

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	DGPSTXID 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	mode	RTCM	2	See *Table 50, DGPS Type* on *page 248*	Enum	4	H
		RTCA	3				
		CMR	4				
		RTCMV3	14				
		AUTO	27				
3	base station ID	Char[5]		ID String See *Table 50, DGPS Type* on *page 248*	Char[5]	8	H+4

2.4.33 DHCPCONFIG
Configure optional parameters for DHCP Server

OEM Platform: ProPak6

Allows users to configure optional parameters for the DHCP Server. The DHCP Server is currently configured to only respond to DHCP requests on the Wi-Fi Interface so the optional PhysicalInterfaceEnum parameter defaults to, and only accepts, "WIFI".

This command is SAVECONFIG-able.

Abbreviated ASCII Syntax: **Message ID: 1710**

dhcpconfig physicalinterfaceenum [enableenum] starting machine number lease time

Factory Default:

dhcp enabled

ASCII Example:

dhcpconfig enable 100 3600

Examples:

dhcpconfig disable

- disables the DHCP Server on the Wi-Fi Interface

dhcpconfig enable

- enables the DHCP Server on the Wi-Fi Interface with default values

dhcpconfig enable 40 259200

- enables the DHCP Server on the Wi-Fi Interface with starting address of x.x.x.40 and lease time of 3 days

Field#	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	DHCPCONFIG Header	Command Header		H	0
2	PhysicalInterfaceEnum	Interface that's being configured for DHCP Default = WIFI (only option)	Enum	4	H + 4
3	EnableEnum	Enable/Disable the DHCP Server Default = Enable	Enum	4	H + 8
4	ULONG	Starting Machine Number part of the IP Address x.x.x.<start> Default = 100 (ie: x.x.x.100)	Ulong	4	H + 12

Field#	Field Type	Data Description	Format	Binary Bytes	Binary Offset
5	ULONG	Lease Time (seconds) 0 = Infinite. Otherwise minimum = 120s Default = 3600 (1 hour)	Ulong	4	H + 16

2.4.34 DIFFCODEBIASCONTROL
Enables /disables satellite differential code biases

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The purpose of the differential code biases is to correct pseudorange errors that affect the L1/L2 ionospheric corrections. This command enables or disables the biases. A set of biases is included in the firmware and use of the biases is enabled by default. See also the `SETDIFFCODEBIASES` command on *page 293.*

Abbreviated ASCII Syntax: **Message ID: 913**

DIFFCODEBIASCONTROL switch

Factory Default:

diffcodebiascontrol enable

Example:

diffcodebiascontrol disable

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	DIFFCODE-BIASCONTROL 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	switch	DISABLE	0	Disable the differential code bias	Enum	4	H
		ENABLE	1	Enable the differential code bias			

2.4.35 DLLTIMECONST
Sets carrier smoothing

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command replaces the GLOCSMOOTH and CSMOOTH commands. It sets the amount of carrier smoothing performed on the code measurements. An input value of 100 corresponds to approximately 100 seconds of smoothing. Upon issuing the command, the locktime (amount of continuous tracking in seconds) for all tracking satellites is reset to zero and each code smoothing filter is restarted. You must wait for at least the length of smoothing time for the new smoothing constant to take full effect. The optimum setting for this command depends on your application.

1. This command may not be suitable for every GNSS application.

2. When using DLLTIMECONST in differential mode with the same receivers, the same setting should be used at both the base and rover station. If the base and rover stations use different types of receivers, it is recommended that you use the command default value is at each receiver (DLLTIMECONST <signaltype> 100).

3. There are several considerations when using the DLLTIMECONST command:

    • The attenuation of low frequency noise (multipath) in pseudorange measurements
    • The effect of time constants on the correlation of phase and code observations
    • The rate of "pulling-in" of the code tracking loop (step response)
    • The effect of ionospheric divergence on carrier smoothed pseudorange (ramp response)

The primary reason for applying carrier smoothing to the measured pseudoranges is to mitigate the high frequency noise inherent in all code measurements. Adding more carrier smoothing by increasing the DLLTIMECONST value filters out lower frequency noise, including some multipath frequencies.

There are also some adverse effects of higher DLLTIMECONST values on some performance aspects of the receiver. Specifically, the time constant of the tracking loop is directly proportional to the DLLTIMECONST value and affects the degree of dependence between the carrier phase and pseudorange information. Carrier phase smoothing of the code measurements (pseudoranges) is accomplished by introducing data from the carrier tracking loops into the code tracking system. Phase and code data, collected at a sampling rate greater than about 3 time constants of the loop, are correlated (the greater the sampling rate, the greater the correlation). This correlation is not relevant if only positions are logged from the receiver, but is an important consideration if the data is combined in some other process such as post-mission carrier smoothing. Also, a narrow bandwidth in a feedback loop impedes the ability of the loop to track step functions. Steps in the pseudorange are encountered during initial lock-on of the satellite and when working in an environment conductive to multipath. A low DLLTIMECONST value allows the receiver to effectively adapt to these situations.

(i) Also, increased carrier smoothing may cause problems when satellite signals are strongly affected by the ionosphere. The rate of divergence between the pseudoranges and phase-derived ranges is greatest when a satellite is low in the sky since the GPS signal must travel through a much "thicker" ionosphere. The tracking error of the receiver is greatest at these times when a lot of carrier smoothing is implemented. In addition, changing periods of ionospheric activity (diurnal changes and the 11-year cycle) influences the impact of large DLLTIMECONST values. It is important to realize that the advantages of carrier smoothing do not come without some trade off in receiver performance. The factory default DLLTIMECONST value of 100 was selected as an optimal compromise of the above considerations. For the majority of applications, this default value should be appropriate. However, the flexibility exists to adjust the parameter for specific applications by users who are familiar with the consequences.

Abbreviated ASCII Syntax: **Message ID: 1011**

DLLTIMECONST signaltype timeconst

Factory Defaults:

dlltimeconst <signaltype>100

Example:

dlltimeconst gpsl2c 100

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	DLLTIMECONST 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	signal type	See *Table 25, Signal Type* on *page 130*	Signal type	Enum	4	H
3	time const		Time constant (sec)	Ulong	4	H+4

Table 25: Signal Type

Value (Binary)	Signal (ASCII)	Description
33	GPSL1CA	GPS L1 C/A-code
68	GPSL2Y	GPS L2 P(Y)-code
69	GPSL2C	GPS L2 C/A-code
70	GPSL2P	GPS L2 P-code
103	GPSL5	GPS L5

Value (Binary)	Signal (ASCII)	Description
2177	GLOL1CA	GLONASS L1 C/A-code
2211	GLOL2CA	GLONASS L2 C/A-code
2212	GLOL2P	GLONASS L2 P-code
12673	BDSB1D1	BeiDou B1 with D1 navigation data
12674	BDSB1D2	BeiDou B1 with D2 navigation data
12803	BDSB2D1	BeiDou B2 with D1 navigation data
12804	BDSB2D2	BeiDou B2 with D2 navigation data
4129	SBASL1	SBAS L1
4194	SBASL5	SBAS L5
10433	GALE1	Galileo E1
10438	GALE1TEST	Galileo E1 test
10466	GALE5A	Galileo E5A
10471	GALE5ATEST	Galileo E5A test
10499	GALE5B	Galileo E5B
10504	GALE5BTEST	Galileo E5B test
10532	GALALTBOC	Galileo ALT-BOC
10537	GALALTBOCTEST	Galileo ALT-BOC test
14753	QZSSL1CA	QZSS L1 C/A-code
14787	QZSSL2C	QZSS L2 C/A-code
14820	QZSSL5	QZSS L5

2.4.36 DNSCONFIG
Manually configures Ethernet DNS servers

OEM Platform: 628, 638, FlexPak6, ProPak6

This command is part of the Ethernet set up. It is used to configure the Domain Name Servers *(*DNS*)* so that host names can be used instead of IP addresses.

> The DNSCONFIG command configures a DNS server for the Ethernet interface, ETHA. However, the receiver will use this DNS server only if the Ethernet interface is configured as the preferred network interface (see SETPREFERREDNETIF on *page 299*).
>
> To configure the DNS server for the Wi-Fi client interface (ProPak6 only), use the WIFICLICONFIG command (see *page 352*).

> The DNSCONFIG command will fail if the IP address for the Ethernet interface, ETHA, is configured to use DHCP. Ensure the IP address for the Ethernet interface is configured to use a static IP address before entering the DNSCONFIG command.
>
> When using DHCP, the DNS server received using DHCP is used and the DNS server configured by DNSCONFIG is ignored.

Abbreviated ASCII Syntax: **Message ID: 1244**

DNSCONFIG number_of_dns_servers ip

Factory Default:

dnsconfig 0

ASCII Example:

dnsconfig 1 192.168.1.5

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	DNSCONFIG 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	NumDNSServer		0	Number of DNS servers	Enum	4	H
		1	1				
3	IP address	ddd.ddd.ddd.ddd		IP address of primary DNS server	String [16]	variable[a]	H+4

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.37 DOSCMD
Issues DOS commands

OEM Platform: 638, ProPak6

Use the DOSCMD command to issue DOS commands to the file system. These commands operate on files and directories located on the internal flash storage file system only. Other than copying files to a USB drive, this command cannot manipulate files on a USB memory stick connected to the OEM638 receiver card.

Abbreviated ASCII Syntax: **Message ID: 1355**

DOSCMD doscommand [string1] [string2]

ASCII Example:

doscmd mkdir survey2	creates a directory named survey2
doscmd copy flightpath.dat	copies a file named flightpath.dat to the external usb memory stick
doscmd del flightpath.dat	deletes a file named flightpath.dat
doscmd del *	deletes all files in the current directory (does not delete directories)

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	DOSCMD header	Command header		H	0
2	DOSCommand	DOS command to issue (see *Table 26, DOS Command Enum* on *page 135*)	Enum	4	H
3	String1	First parameter for the command (see *Table 26, DOS Command Enum* on *page 135*) (optional)	Char (null terminated)	128	variable
4	String2	Second parameter for the command (see *Table 26, DOS Command Enum* on *page 135*) (optional)	Char (null terminated)	128	variable

ⓘ   For a list of the files and directories on the file system, use the DIRENT log (see *page 443*).

Table 26: DOS Command Enum

Binary Value	ASCII Value	Description	String 1	String 2	Comments
2	CD	Change directory	Directory name		
3	FORMAT[a]	Format media			Low level disk initialization. Resets the receiver when complete
4	RMDIR	Remove directory	Directory name		Directory must be empty
5	MKDIR	Make directory	Directory name		
7	COPY	Copy file to USB	File name		USB stick must be formatted and inserted
8	DEL	Delete file	File name		* wildcard symbol deletes all files in current directory
9	RENAME	Rename file	Old name	New name	

a. The FORMAT command prepares a new receiver and can take 2 minutes or more to complete. This command erases all data and cannot be recovered.

2.4.38   DUALANTENNAALIGN
Dual Antenna ALIGN configuration

OEM Platform: 617D, FlexPak6D

This command can be used to enable or disable ALIGN and configure the ALIGN operation rates on a dual antenna OEM617D board and FlexPak6D enclosure.

Abbreviated ASCII Syntax: **Message ID: 1761**

dualantennaalign switch [obsrate] [posrate]

Factory Default:

dualantennaalign enable 1 1

Example:

dualantennaalign enable 10 1

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	DUALANTENNA ALIGN 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	Switch	DISABLE	0	Disable the dual antenna ALIGN feature	Enum	4	H
		ENABLE	1	Enable the dual antenna ALIGN feature			
3	obsrate	1, 2, 4, 5, 10 or 20 Hz[a]		Rate at which heading output is required (default 1 Hz)	Ulong	4	H+4
4	posrate	1, 2, 4, 5, 10 or 20 Hz**[a]**		Rate at which MASTERPOS output is required (default 1 Hz)	Ulong	4	H+8

a. Dual antenna ALIGN rates are limited to the maximum position rate allowed by the receiver model.

2.4.39 DUALANTENNAPOWER
Controls power to the secondary antenna

OEM Platform: ProPak6

Use the `DUALANTENNAPOWER` command to control the LNA power to the secondary GNSS antenna.

> (i) This command is only applicable to ProPak6 receivers with the Heading option.

> (i) This command controls only the LNA for the secondary GNSS antenna. The primary GNSS antenna is controlled by the ANTENNAPOWER command (see *page 70*).

Abbreviated ASCII Syntax: **Message ID: 1639**

DUALANTENNAPOWER antennapower

Factory Default:

dualantennapower on

ASCII Example:

dualantennapower off

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	DUALANTENNA POWER 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	antennapower	OFF	0	Power off immediately	Enum	4	H
		ON	1	Power on immediately			

2.4.40   DYNAMICS
Tunes receiver parameters

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to adjust the receiver dynamics to that of an application. It is used to optimally tune receiver parameters.

The DYNAMICS command adjusts the Tracking State transition time out value of the receiver, see *Table 123, Tracking State* on *page 602*. When the receiver loses the position solution, see *Table 85, Solution Status* on *page 408*, it attempts to steer the tracking loops for fast reacquisition (5 s time-out by default). The DYNAMICS command adjusts this time-out value, effectively increasing the steering time. The three states AIR, LAND or FOOT set the time-out to 5, 10 or 20 seconds respectively.

> (i) The DYNAMICS command should only be used by advanced users. The default of AUTO should **not** be changed except under very specific conditions.

Abbreviated ASCII Syntax: **Message ID: 258**

DYNAMICS settings

Factory Default:

  dynamics auto

Example:

  dynamics foot

Table 27: User Dynamics

Binary	ASCII	Description
0	AIR	Receiver is in an aircraft or a land vehicle, for example a high speed train, with velocity greater than 110 km/h (30 m/s). This is also the most suitable dynamic for a jittery vehicle at any speed.
1	LAND	Receiver is in a stable land vehicle with velocity less than 110 km/h (30 m/s).
2	FOOT	Receiver is being carried by a person with velocity less than 11 km/h (3 m/s).
3	AUTO	Receiver monitors dynamics and adapts behavior accordingly

> (i) Qualifying North American Solar Challenge cars annually weave their way through 1000's of miles between the US and Canada. GNSS keeps them on track through many intersections on secondary highways and gives the Calgary team constant intelligence on the competition's every move. In this case, with average speeds of 46 miles/hour and at times a jittery vehicle, AIR is the most suitable dynamic.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	DYNAMICS 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	settings	See *Table 27, User Dynamics* on *page 138*		Receiver dynamics based on the current environment	Enum	4	H

2.4.41   ECHO
Sets port echo

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set a port to echo.

Abbreviated ASCII Syntax: **Message ID: 1247**

ECHO [port] echo

Factory Default:

echo com1 off	
echo com2 off	
echo com3 off	
echo com4 off	(OEM638 only)
echo com5 off	(OEM638 only)
echo com6 off	(OEM638 only)
echo usb1 off	
echo usb2 off	
echo usb3 off	
echo icom1 off	(not supported on OEM615)
echo icom2 off	(not supported on OEM615)
echo icom3 off	(not supported on OEM615)

ASCII Example:

echo com1 on
echo on

Field #	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	ECHO 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	See *Table 28, Communications Port Identifiers* on *page 141*[a]		Port to configure (default = THISPORT)	Enum	4	H
3	echo	OFF	0	Sets port echo to off	Enum	4	H+4
		ON	1	Sets port echo to on			

a. XCOM and NCOM ports are not supported.

Table 28: Communications Port Identifiers

ASCII Port Name	Binary Value
NOPORT	0
COM1	1
COM2	2
COM3	3
THISPORT	6
FILE	7
XCOM1	9
XCOM2	10
USB1	13
USB2	14
USB3	15
XCOM3	17
COM4	19
IMU	21
ICOM1	23
ICOM2	24
ICOM3	25
NCOM1	26
NCOM2	27
NCOM3	28
WCOM1	30
COM5	31
COM6	32
BT1	33
COM7	34
COM8	35
COM9	36
COM10	37

2.4.42   ECUTOFF
Sets satellite elevation cut-off for GPS Satellites

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the elevation cut-off angle for tracked GPS satellites. The receiver does not start automatically searching for a GPS satellite until it rises above the cut-off angle (when satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they are manually assigned (see the ASSIGN command *Section 2.4.6* on *page 72*).

In either case, satellites below the ECUTOFF angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle; it could be used in these situations:

- The antenna is at a high altitude, and thus can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction

> 1. Care must be taken when using `ECUTAOFF` command because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.
>
> 2. Use the `ELEVATIONCUTOFF` command (*page 144*) to set the cut-off angle for all other systems.
>
> 3. For the OEM617D and FlexPak6D receivers, this command applies to both the primary and secondary antennas.

Abbreviated ASCII Syntax: **Message ID: 50**

ECUTOFF angle

Factory Default:

ecutoff 5.0

ASCII Example:

ecutoff 10.0

> ⓘ A low elevation satellite is a satellite the receiver tracks just above the horizon. Generally, a satellite is considered low elevation if it is between 0 and 15 degrees above the horizon.
>
> There is no difference between the data transmitted from a low elevation satellite and that transmitted from a higher elevation satellite. However, differences in the signal path of a low elevation satellite make their use less desirable. Low elevation satellite signals have more error due to the increased amount of atmosphere they must travel through. In addition, signals from low elevation satellites don't fit the assumption that a signal travels in air nearly the same as in a vacuum. As such, using low elevation satellites in the solution results in greater position inaccuracies.
>
> The elevation cut-off angle is specified with ECUTOFF to ensure that noisy, low elevation satellite data below the cut-off is not used in computing a position. If post-processing data, it is still best to collect all data (even that below the cut-off angle). Experimenting with different cut-off angles can then be done to provide the best results. In cases where there are not enough satellites visible, a low elevation satellite may actually help in providing a useful solution.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	ECUTOFF 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	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

2.4.43   ELEVATIONCUTOFF
Sets the elevation cut-off angle for tracked satellites

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The `ELEVATIONCUTOFF` command is used to set the elevation cut-off angle for tracked satellites. The receiver does not start automatically searching for a satellite until it rises above the cut-off angle (when the satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they are manually assigned (refer to the `ASSIGN` command on *page 72*).

In either case, satellites below the elevation cut-off angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle and can be used in the following situations:

- The antenna is at a high altitude and thus can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction

> 1. Care must be taken when using `ELEVATIONCUTOFF` command because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.
>
> 2. This command combines the following commands into one convenient command: `ECUTOFF, GLOECUTOFF, GALECUTOFF, QZSSECUTOFF, SBASECUTOFF` and `BDSECUTOFF`.
>
> 3. For the OEM617D and FlexPak6D receivers, this command applies to both the primary and secondary antennas.

A low elevation satellite is a satellite the receiver tracks just above the horizon. Generally, a satellite is considered low elevation if it is between 0 and 15 degrees above the horizon.

There is no difference between the data transmitted from a low elevation satellite and that transmitted from a higher elevation satellite. However, differences in the signal path of a low elevation satellite make their use less desirable. Low elevation satellite signals have more error due to the increased amount of atmosphere they must travel through. In addition, signals from low elevation satellites don't fit the assumption that a signal travels in air nearly the same as in a vacuum. As such, using low elevation satellites in the solution results in greater position inaccuracies.

The elevation cut-off angle is specified with the `ELEVATIONCUTOFF` command to ensure that noisy, low elevation satellite data below the cut-off is not used in computing a position. If post-processing data, it is still best to collect all data (even that below the cutoff angle). Experimenting with different cut-off angles can then be done to provide the best results. In cases where there are not enough satellites visible, a low elevation satellite may actually help in providing a useful solution.

Abbreviated ASCII Syntax: **Message ID: 1735**

elevationcutoff constellation angle [reserved]

Factory default:

elevationcutoff all 5.0 0

ASCII Example:

elevationcutoff gps 5

elevationcutoff all 5

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	ELEVATION CUTOFF 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	Constellation	GPS	0	Sets the cut-off angle for GPS Constellation satellites only.	Enum	4	H
		GLONASS	1	Sets the cut-off angle for GLONASS constellation satellites only			
		SBAS	2	Sets the cut-off angle for SBAS constellation satellites only			
		GALILEO	5	Sets the cut-off angle for Galileo constellation satellites only			
		BEIDOU	6	Sets the cut-off angle for BeiDou constellation satellites only			
		QZSS	7	Sets the cut-off angle for QZSS constellation satellites only			
		NONE	31				
		ALL	32	Sets the cut-off angle for all satellites regardless of the constellation.			
3	Angle	±90.0 degrees		Elevation cut-off angle relative to the horizon.	Float	4	H+4
4	Reserved	0		Reserved Field (optional)	Ulong	4	H+8

2.4.44 ETHCONFIG
Configures Ethernet physical layer

OEM Platform: 628, 638, FlexPak6, ProPak6

This command is used to configure the Ethernet physical layer.

Abbreviated ASCII Syntax: **Message ID: 1245**

ETHCONFIG interface_name [speed] [duplex] [crossover] [power_mode]

Factory Default:

> ethconfig etha auto auto auto powerdown

ASCII Example:

> ethconfig etha 100 full mdix normal

Field #	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	ETHCONFIG 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	interface_name	ETHA	2		Enum	4	H
3	speed[a][b]	AUTO	1	Auto-negotiate speed (default)	Enum	4	H+4
		10	2	Force 10BaseT			
		100	3	Force 100BaseT			
4	duplex[b]	AUTO	1	Auto-negotiate duplex (default)	Enum	4	H+8
		HALF	2	Force half duplex			
		FULL	3	Force full duplex			
5	crossover[a]	AUTO	1	Auto-detect crossover (default)	Enum	4	H+12
		MDI	2	Force MDI (straight through)			
		MDIX	3	Force MDIX (crossover)			
6	power_mode	AUTO	1	Energy detect mode	Enum	4	H+16
		POWER DOWN	2	Soft power down mode (default)			
		NORMAL	3	Normal mode			

a. AUTO is the recommended value for field types Speed and Crossover.
b. If setting Speed and/or Duplex to AUTO, both must be set to AUTO at the same time otherwise a "parameter 3 out of range" error occurs.

2.4.45 EVENTINCONTROL
Controls Event-In input triggers

OEM Platform: 638, ProPak6

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 (refer to the MARKxTIME or MARKxPOS logs in 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 an Event Input Trigger, it is possible to overwhelm the receiver with a very high rate of input events that impacts the performance of the receiver. For this reason, the receiver internally throttles the rate at which it responds to input events. The limit is 200 Hz. Throttling only applies when the input is used as an event strobe input; throttling does not apply when used in pulse counter mode.

When used as a pulse counter, an internal accumulator is used to increment each input pulse and output each second using the MARKCOUNT log coordinated with 1 PPS. 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 *page 188*).

Abbreviated ASCII Syntax: **Message ID: 1637**

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

ASCII Example:

> eventincontrol mark1 enable

Field	Field Type	ASCII Value	Binary Value	Description	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, respectively	-	H	0
2	mark	MARK1	0	Choose which Event-In Mark to change. This value must be specified.	Enum	4	H
		MARK2	1				
		MARK3	2				
		MARK4	3				

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	switch	DISABLE	0	Disables Event Input	Enum	4	H+4
		EVENT	1	Enables Event Input (default)			
		COUNT	2	Increments a counter with each input pulse (for example, a wheel sensor. The period of the count is from 1 PPS to the next PPS.			
		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		Time bias in nanoseconds If Field 3 is COUNT, this field is not used.	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. If Field 3 is COUNT, this field is not used.	Ulong	4	H+16

2.4.46 EVENTOUTCONTROL
Control Event-Out properties

OEM Platform: 638, ProPak6

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.

> (i) 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 *page 188*).
> On the OEM638, MARK1 through MARK7 are available. On the ProPak6 only MARK1 through MARK3 and MARK7 (IMU_EVENT_OUT) are available on the Personality Port.

Abbreviated ASCII Syntax: **Message ID: 1636**

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

ASCII Example:

eventoutcontrol mark3 enable

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	EVENTOUTC ONTROL 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	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)			

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
4	polarity	NEGATIVE	0	Negative polarity (active = 0V) (default)	Enum	4	H+8
		POSITIVE	1	Positive polarity (active = 3.3V)			
5	active_period[a]	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**[a]**	default: 500,000,000 minimum: 10 maximum: 999,999,990		Non-active period of the Event Out signal in nanoseconds	Ulong	4	H+16

a. The sum of the active period and inactive period should total 1,000,000,000ns. However, this is not enforced. If the total exceeds 1 full second, the active period duration will be as given and the inactive period will be the remainder of the second. If the sum is less than 1 full second an unpredictable periodic output will result.

2.4.47   EXTERNALCLOCK
Sets external clock parameters

OEM Platform: 628, 638, ProPak6

Overview

The EXTERNALCLOCK command is used to enable the OEM6 card to operate with an optional external oscillator. You are able to optimally adjust the clock model parameters of these receivers for various types of external clocks.

> (i)   1.   This command affects the interpretation of the CLOCKMODEL log.
>
>   2.   If the EXTERNALCLOCK command is enabled and set for an external clock (TCXO, OCXO, RUBIDIUM, CESIUM or USER) and the CLOCKAD-JUST command, see *page 102*, is ENABLED, then the clock steering process takes over the VARF output pins and may conflict with a previously entered FREQUENCYOUT command, see *page 161*. If clocksteering is not used with the external oscillator, the clocksteering process must be disabled by using the CLOCKADJUST DISABLE command.
>
>   3.   When using the EXTERNALCLOCK and CLOCKADJUST commands together, issue the EXTERNALCLOCK command first to avoid losing satellites.

There are three steps involved in using an external oscillator:

1.   Follow the procedure outlined in the OEM6 Family Installation and Operation User Manual (OM-20000128) to connect an external oscillator to the OEM6.

2.   Using the EXTERNALCLOCK command, select a standard oscillator and its operating frequency.

3.   Using the CLOCKADJUST command, disable the clocksteering process if external clocksteering is not used.

Theory

An unsteered oscillator can be approximated by a three-state clock model, with two states representing the range bias and range bias rate, and a third state assumed to be a Gauss-Markov (GM) process representing the range bias error generated from satellite clock dither. The third state is included because the Kalman filter assumes an (unmodeled) white input error. The significant correlated errors produced by satellite clock dither are obviously not white and the Markov process is an attempt to handle this kind of short term variation.

The internal units of the new clock model's three states (offset, drift and GM state) are metres, metres per second and metres. When scaled to time units for the output log, these become seconds, seconds per second and seconds, respectively.

The user has control over 3 process noise elements of the linear portion of the clock model. These are the h_0, h_{-1} and h_{-2} elements of the power law spectral density model used to describe the frequency noise characteristics of oscillators:

$$S_y(f) = \frac{h_{-2}}{f^2} + \frac{h_{-1}}{f} + h_0 + h_1 f + h_2 f^2$$

where *f* is the sampling frequency and $S_y(f)$ is the clock's power spectrum. Typically only h_0, h_{-1}, and h_{-2} affect the clock's Allan variance and the clock model's process noise elements.

Usage

Before you use an optional external oscillator, several clock model parameters must be set. There are default settings for a Voltage-Controlled Temperature-Compensated Crystal Oscillator (VCTCXO), Ovenized Crystal Oscillator (OCXO), Rubidium and Cesium standard, which are given in *Table 29, Page 153.* You may alternatively choose to supply customized settings.

> ⓘ The EXTERNALCLOCK command determines whether the receiver uses its own internal temperature-compensated crystal oscillator or that of an external oscillator as a frequency reference. It also sets which clock model is used for an external oscillator.
>
> To force the OEM6 to use the internal oscillator, use the EXTERNALCLOCK DISABLE command and physically disconnect the external oscillator input. Do not use the EXTERNALCLOCK OCXO, CESIUM, RUBIDIUM or USER parameters if there is no external oscillator connected to the OEM6.

Abbreviated ASCII Syntax: **Message ID: 230**

EXTERNALCLOCK clocktype [freq] [h_0[h_{-1}[h_{-2}]]]

Factory Default:

externalclock disable

ASCII Examples:

externalclock user 10mhz 1.0167e-23 6.87621e-25 8.1762e-26
externalclock tcxo 5mhz

Table 29: Clock Type

ASCII	Binary	Description
DISABLE	0	Turns the external clock input off, reverts back to the on-board VCTCXO
TCXO	1	Sets the predefined values for a VCTCXO
OCXO	2	Sets the predefined values for an OCXO
RUBIDIUM	3	Sets the predefined values for a rubidium oscillator
CESIUM	4	Sets the predefined values for a cesium oscillator
USER	5	Defines custom process noise elements

Table 30: Pre-Defined Values for Oscillators

Clock Type	h_0	h_{-1}	h_{-2}
VCTCXO	1.0 e-21	1.0 e-20	1.0 e-20
OCXO	2.51 e-26	2.51 e-23	2.51 e-22
Rubidium	1.0 e-23	1.0 e-22	1.3 e-26
Cesium	2.0 e-20	7.0 e-23	4.0 e-29

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	EXTERNALCLOCK 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	clocktype	See *Table 29, Page 153*		Clock type	Enum	4	H
3	freq	0MHz	0	Optional frequency. If a value is not specified, the default is 5 MHz	Enum	4	H+4
		5MHz	1				
		10MHz	2				
4	h_0	1.0 e-35 to 1.0 e-18		Optional timing standards. These fields are only valid when the USER clocktype is selected. Do not use h values with VCTCXO, OCXO, CESIUM or RUBIDIUM clock types. The h values for these options are fixed, see *Table 30* (default=0.0)	Double	8	H+8
5	h_{-1}	1.0 e-35 to 1.0 e-18			Double	8	H+16
6	h_{-2}	1.0 e-35 to 1.0 e-18			Double	8	H+24

2.4.48 FIX
Constrains to fixed height or position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to fix height or position to the input values. For various applications, fixing these values can assist in improving acquisition times and accuracy of position or corrections. For example, fixing the position is a requirement for differential base stations as it provides the reference position to base the differential corrections from.

If you enter a FIXPOSDATUM command, see *page 158*, the FIX command is then issued internally with the FIXPOSDATUM command values translated to WGS84. It is the FIX command that appears in the RXCONFIG log. If the FIX or the FIXPOSDATUM command are used, their newest values overwrite the internal FIX values.

> ⓘ
> 1. It is strongly recommended that the FIX POSITION entered be accurate to within a few metres. This level of accuracy can be obtained from a receiver using single point positioning once 5 or 6 satellites are being tracked.
>
> 2. FIX POSITION should only be used for base station receivers. Applying FIX POSITION to a rover switches it from RT2 mode to a fixed position mode. Applying FIX POSITION to the rover does not speed up ambiguity resolution.
>
> 3. Any setting other than FIX POSITION disables output of differential corrections unless the MOVINGBASESTATION command is set to ENABLE, see *page 217*.
>
> 4. You can fix the position of the receiver using latitude, longitude and height in Mean Sea Level (MSL) or ellipsoidal parameters depending on the UNDULATION setting. The factory default for the UNDULATION (*page 335*) setting is EGM96, where the height entered in the FIX command is set as MSL height. If you change the UNDULATION setting to USER 0, the height entered in the FIX command is set as ellipsoidal height (refer to *Table 31)*.

Error checking is performed on the entered fixed position by the integrity monitor. Depending on the result of this check, the position can be flagged with the following statuses.

SOL_COMPUTED: The entered position has been confirmed by measurement.

PENDING: Insufficient measurements are available to confirm the entered position.

INTEGRITY_WARNING: First level of error when an incorrect position has been entered. The fixed position is off by approximately 25-50 meters.

INVALID_FIX: Second level of error when an inaccurate position has been entered. The fixed position is off by a gross amount.

> ⓘ An incorrectly entered fixed position will be flagged either
> INTEGRITY_WARNING or INVALID_FIX. This will stop output of differential
> corrections or RTK measurements and can affect the clock steering and
> satellite signal search. Checks on the entered fixed position can be disabled
> using the `RAIMMODE` command (*page 252*).

Abbreviated ASCII Syntax: **Message ID: 44**

FIX type [param1 [param2 [param3]]]

Factory Default:

       fix none

ASCII Example:

       fix none
       fix height 4.567
       fix position 51.116 -114.038 1065.0

> ⓘ In order to maximize the absolute accuracy of RTK rover positions, the base
> station coordinates must be fixed to their known position using the `FIX`
> `POSITION` [lat][lon][hgt] command.

Table 31: FIX Parameters

ASCII Type Name	Parameter 1	Parameter 2	Parameter 3
AUTO	Not used	Not used	Not used
HEIGHT	Default MSL height [a] [b] (-1000 to 20000000 m)	Not used	Not used
NONE	Not used	Not used	Not used
POSITION	Lat (-90 to 90 degrees) where a '-' sign denotes south and a '+' sign denotes north	Lon (-360 to 360 degrees) where a '-' sign denotes west and a '+' sign denotes east	Default MSL height [a] [b] (-1000 to 20000000 m)

a. For a discussion on height, refer to our An Introduction to GNSS book, available on our website at http://www.novatel.com/support/knowledge-and-learning/.
b. See also Note #4 on the previous page.

Table 32: Fix Types

ASCII Name	Binary Value	Description
NONE	0	Unfix. Clears any previous FIX commands
AUTO	1	Configures the receiver to fix the height at the last calculated value if the number of satellites available is insufficient for a 3-D solution. This provides a 2-D solution. Height calculation resumes when the number of satellites available allows a 3-D solution
HEIGHT	2	Configures the receiver in 2-D mode with its height constrained to a given value. This command is used mainly in marine applications where height in relation to mean sea level may be considered to be approximately constant. The height entered using this command is referenced to the mean sea level, see the BESTPOS log on *page 406* (is in metres). The receiver is capable of receiving and applying differential corrections from a base station while FIX HEIGHT is in effect. The FIX HEIGHT command overrides any previous `FIX HEIGHT` or `FIX POSITION` command. ⓘ This command only affects pseudorange corrections and solutions.
POSITION	3	Configures the receiver with its position fixed. This command is used when it is necessary to generate differential corrections. For both pseudorange and differential corrections, this command must be properly initialized before the receiver can operate as a GNSS base station. Once initialized, the receiver computes differential corrections for each satellite being tracked. The computed differential corrections can then be output to rover stations by utilizing any of the following receiver differential corrections data log formats: RTCM, RTCMV3, RTCA or CMR. See the OEM6 Family Installation and Operation User Manual (OM-20000128) for information about using the receiver for differential applications. The values entered into the FIX POSITION command should reflect the precise position of the base station antenna phase center. Any errors in the FIX POSITION coordinates directly bias the corrections calculated by the base receiver. The receiver performs all internal computations based on WGS84 and the datum command is defaulted as such. The datum in which you choose to operate (by changing the `datum` command) is internally converted to and from WGS84. Therefore, all differential corrections are based on WGS84, regardless of your operating datum. The `FIX POSITION` command overrides any previous `FIX HEIGHT` or `FIX POSITION` command settings.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	FIX 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	type	See *Table 32, Page 156*		Fix type	Enum	4	H
3	param1	See *Table 31, Page 155*		Parameter 1	Double	8	H + 4
4	param2			Parameter 2	Double	8	H + 12
5	param3			Parameter 3	Double	8	H + 20

2.4.49   FIXPOSDATUM
Sets position in a specified datum

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the FIX position in a specific datum. The input position is transformed into the same datum as that in the receiver's current setting. The `FIX` command, see *page 154*, is then issued internally with the `FIXPOSDATUM` command values. It is the `FIX` command that appears in the RXCONFIG log. If the `FIX` or the `FIXPOSDATUM` command are used, their newest values overwrite the internal FIX values.

Abbreviated ASCII Syntax: **Message ID: 761**

FIXPOSDATUM datum lat lon height

Factory Default:

> fix none

ASCII Example:

> fixposdatum user 51.11633810554 -114.03839550586 1048.2343

> Use the `FIXPOSDATUM` command in a survey to fix the position with values from another known datum, rather than manually transforming them into WGS84.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	FIXPOSDATUM 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	datum	See *Table 23, Reference Ellipsoid Constants* on *page 119*	Datum ID	Enum	4	H
3	lat	±90		Latitude (degrees)	Double	8	H + 4
4	lon	±360		Longitude (degrees)	Double	8	H + 12
5	height	-1000 to 20000000		Mean sea level (MSL) height (m) [a]	Double	8	H + 20

a. For a discussion on height, refer to our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

2.4.50 *FORCEGLOL2CODE*
Forces receiver to track GLONASS satellite L2 P or L2 C/A code

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to force the receiver to track GLONASS satellite L2 P-code or L2 C/A code.

Abbreviated ASCII Syntax: **Message ID: 1217**

FORCEGLOL2CODE l2type

Factory Default:

forceglol2code default

ASCII Example:

forceglol2code p

Table 33: GLONASS L2 Code Type

Binary	ASCII	Description
1	P	L2 P-code or L2 Precise code
2	C	L2 C/A code or L2 Coarse/Acquisition code
3	DEFAULT	Set to channel default

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	FORCEGLOL2-CODE 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	L2type	See *Table 33* above		GLONASS L2 code type	Enum	4	H

2.4.51 FORCEGPSL2CODE
Forces receiver to track GPS satellite L2 P or L2C code

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to force the receiver to track GPS L2 P-code or L2C code. AUTO tells the receiver to use L2C code type if available and L2 P-code if L2C code is not available.

Abbreviated ASCII Syntax: **Message ID: 796**

FORCEGPSL2CODE l2type

Factory Default:

> forcegpsl2code default

ASCII Example:

> forcegpsl2code p

Table 34: GPS L2 Code Type

Binary	ASCII	Description
0	AUTO	Receiver uses the L2C if available and L2 P otherwise. An exception is when the receiver is doing RTK positioning. In that case, AUTO changes the L2 code type being tracked to match the L2 code type found in the base station corrections, which ensures the greatest number of satellites are used in the solution.
1	P	L2 P-code or L2 Precise code
2	C	L2C code or L2 Civilian code
3	DEFAULT	Set to channel default

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	FORCEGPSL2-CODE 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	L2type	See *Table 34* above	GPS L2 code type	Enum	4	H

2.4.52 FREQUENCYOUT
Sets output pulse train available on VARF

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the output pulse train available on the Variable Frequency (VARF) pin. The output waveform is coherent with the 1PPS output, see the usage note and Figure 4, Pulse Width and 1PPS Coherency on Page 162.

> (i) 1. If the CLOCKADJUST command is ENABLED (*page 102*) and the receiver is configured to use an external reference frequency (set in the EXTERNALCLOCK command (*page 151*), for an external clock - TCXO, OCXO, RUBIDIUM, CESIUM, or USER), then the clock steering process takes over the VARF output pins and may conflict with a previously entered FREQUENCYOUT command.
>
> 2. *Figure 4* on *Page 162* shows how the chosen pulse width is frequency locked but not necessarily phase locked when using ENABLE option. To synchronize the phase, use ENABLESYNC option.

Abbreviated ASCII Syntax: **Message ID: 232**

FREQUENCYOUT [switch] [pulsewidth] [period]
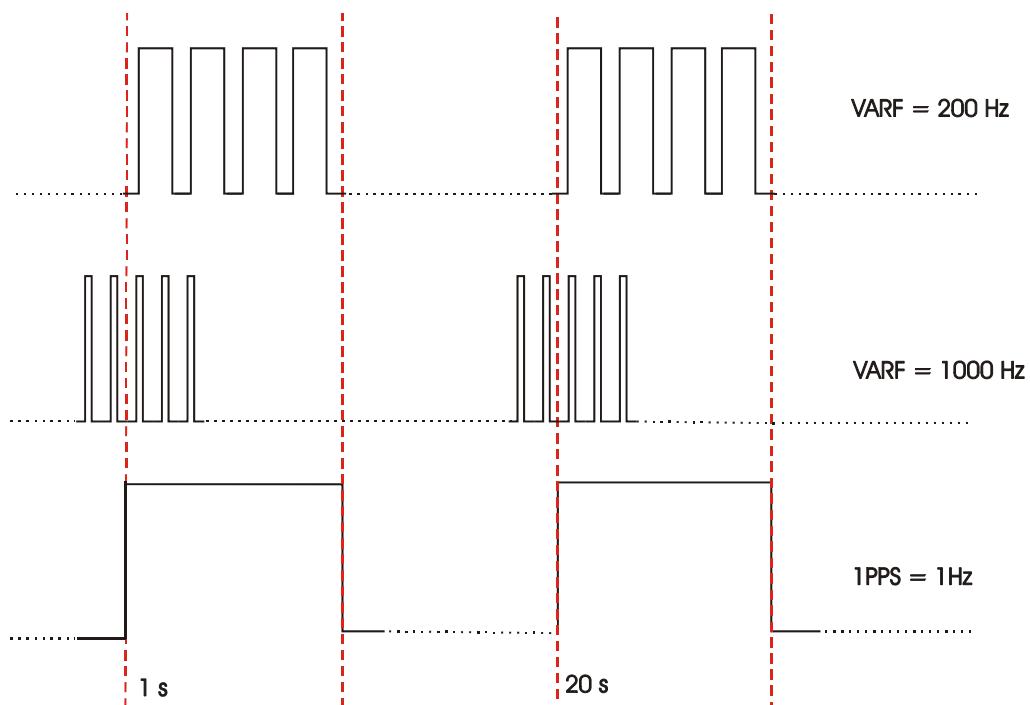
Factory Default:

frequencyout disable

ASCII Example:

frequencyout enable 2 4

This example generates a 50% duty cycle 25 MHz square wave.

Figure 4: Pulse Width and 1PPS Coherency

VARF = 200 Hz

VARF = 1000 Hz

1PPS = 1Hz

1 s

20 s

(i) When using ENABLE option, the VARF and 1PPS are not necessarily in phase as described in the diagram above. To align the phase of the VARF with the 1PPS, use the ENABLESYNC option and the VARF phase will be synchronized to the leading edge of the 1PPS pulse. Note that if the VARF and 1PPS frequencies are not even multiples of each other, this may cause the VARF to have a shorter cycle pulse prior to each 1PPS pulse. 1PPS is not affected.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	FREQUENCYOUT 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	switch	DISABLE	0	Disable causes the output to be fixed low (if NONE specified, defaults to DISABLE)	Enum	4	H
		ENABLE	1	Enables customized frequency output			
		ENABLE SYNC	2	Enable customized frequency output synchronized to PPS			
3	pulsewidth	(0 to 1073741823)		Number of 10 ns steps for which the output is high. Duty cycle = pulsewidth / period. If pulsewidth is greater than or equal to the period, the output is a high DC signal. If pulsewidth is 1/2 the period, then the output is a square wave (default = 0)	Ulong	4	H+4
4	period	(0 to 1073741823)		Signal period in 10 ns steps. Frequency Output = 100,000,000 / Period (default = 0)	Ulong	4	H+8

2.4.53   FRESET
Clears selected data from NVM and reset

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to clear data which is stored in non-volatile memory. Such data includes the almanac, ephemeris, and any user specific configurations. The commands, ephemeris, almanac, and L-Band related data, excluding the subscription information, can be cleared by using the STANDARD target. The model can only be cleared by using the MODEL target. The receiver is forced to hardware reset. In addition, values entered using the CLOCKCALIBRATE command can only be cleared by using the CLKCALIBRATION target.

> ⓘ   FRESET STANDARD (which is also the default) causes most commands, ephemeris, GNSS and almanac data previously saved to NVM to be erased.

Abbreviated ASCII Syntax: **Message ID: 20**

FRESET [target]

Input Example:

> freset command

> ⓘ   If you are receiving no data or random data from your receiver, try the following before contacting NovAtel:
>
> - Verify that the receiver is tracking satellites
> - Check the integrity and connectivity of power and data cables
> - Verify the baud rate settings of the receiver and terminal device (your PC, data logger or laptop)
> - Switch COM ports
>
> Issue the FRESET command

Table 35: FRESET Target

Binary	ASCII	Description
0	STANDARD	Resets commands (except CLOCKCALIBRATION and MODEL), ephemeris and almanac (default). Also resets all L-Band related data except for the subscription information. Does not reset the Ethernet settings.
1	COMMAND	Resets the stored commands (saved configuration)
2	GPSALMANAC	Resets the stored GPS almanac
3	GPSEPHEM	Resets the stored GPS ephemeris
4	GLOEPHEM	Resets the stored GLONASS ephemeris

Binary	ASCII	Description
5	MODEL	Resets the currently selected model
11	CLKCALIBRATION	Resets the parameters entered using the `CLOCKCALIBRATE` command
20	SBASALMANAC	Resets the stored SBAS almanac
21	LAST_POSITION	Resets the position using the last stored position
31	GLOALMANAC	Resets the stored GLONASS almanac
39	GALFNAV_EPH	Resets the stored GALFNAV ephemeris
40	GALINAV_EPH	Resets the stored GALINAV ephemeris
45	GALFNAV_ALM	Resets the stored GALFNAV almanac
46	GALINAV_ALM	Resets the stored GALINAV almanac
52	PROFILEINFO	Resets the stored profile configurations
54	QZSSALMANC	Resets the QZSS almanac
55	QZSSEPHEMERIS	Resets the QZSS ephemeris
57	BDSALMANAC	Resets the BeiDou almanac
58	BDSEPHEMERIS	Resets the BeiDou ephemeris
60	USER_ACCOUNTS	Resets the admin password to the default (the receiver's PSN)
64	ETHERNET	Resets the stored Ethernet settings

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	FRESET 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	target	See *Table 35, FRESET Target* on *page 164*	What data is to be reset by the receiver (default = STANDARD)	Enum	4	H	

2.4.54   GALECUTOFF
Sets elevation cut-off angle for Galileo satellites

OEM Platform: 628, 638, FlexPak6, ProPak6

This command is used to set the elevation cut-off angle for tracked Galileo satellites. The receiver does not start automatically searching for a satellite until it rises above the cut-off angle (when satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they were manually assigned (see the ASSIGN command starting on *page 72*).

In either case, satellites below the GALECUTOFF angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle and can be used in the following situations:

- • The antenna is at a high altitude and thus look below the local horizon
- • Satellites are visible below the horizon due to atmospheric refraction

> ⓘ
> 1. Care must be taken when using GALECUTOFF because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.
>
> 2. Use the ELEVATIONCUTOFF command (*page 144*) to set the cut-off angle for all other systems.

Abbreviated ASCII Syntax: **Message ID: 1114**

GALECUTOFF angle

Factory Default:

       galecutoff 5.0

ASCII Example:

       galecutoff 10.0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	GALECUTOFF 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	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

2.4.55 GENERATEALIGNCORRECTIONS
Configure ALIGN Master

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to configure the ALIGN Master and starts sending out ALIGN RTCA corrections through the specified port. This command is equivalent to sending the following commands to the Master:

> unlogall [port]
> fix none
> movingbasestation enable
> interfacemode [port] novatel rtca
> com [port] [baud] n 8 1 n off on
> log [port] rtcaobs3 ontime [rate = 1/ obsreqrate]
> log [port] rtcarefext ontime [rate = 1/ refextreqrate]

Abbreviated ASCII Syntax: **Message ID: 1349**

GENERATEALIGNCORRECTIONS port [baud] [obsreqrate] [refextreqrate]

ASCII Example:

> generatealigncorrections com2 230400 10 10

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	GENERATE ALIGNCORR ECTIONS 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	See *Table 57, COM Port Identifiers* on *page 284*		Port through which corrections should be sent out	Enum	4	H
3	baud	9600, 19200, 38400, 57600, 115200, 230400, 460800 or 921600		Communication baud rate (bps) (default = 9600)	Ulong	4	H+4
4	obsreqrate	1, 2, 4, 5, 10 or 20		RTCAOBS3 data rate in Hz (default = 1)	Ulong	4	H+8
5	refextreqrate	0, 1, 2, 4, 5, 10 or 20		RTCAREFEXT data rate in Hz (default = 1)	Ulong	4	H+12

2.4.56 GENERATEDIFFCORRECTIONS
Sends a preconfigured set of differential corrections

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to configure the receiver to send a preconfigured set of differential pseudorange corrections.

Abbreviated ASCII Syntax:　　　　　　**Message ID: 1296**

GENERATEDIFFCORRECTIONS mode [port]

ASCII Example:

generatediffcorrections rtcm com2

Preconfigured set of differential corrections sent when RTCM:

rtcm1 ontime 1
rtcm31 ontime 1
rtcm3 ontime 10

Preconfigured set of differential corrections sent when RTCA:

rtca1 ontime 1
rtcaref ontime 10

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	GENERATE DIFFCORRE CTIONS 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	mode	RTCM	2	Serial port interface mode identifier. See *Table 37, Serial Port Interface Modes* on *page 186*	Enum	4	H
		RTCA	3				
3	port	See *Table 57, COM Port Identifiers* on *page 284*		Port to configure (default = THISPORT)	Enum	4	H+4

2.4.57   *GENERATERTKCORRECTIONS*
Sends a preconfigured set of RTK corrections

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to configure the receiver to send a preconfigured set of RTK (carrier phase) corrections.

Abbreviated ASCII Syntax: **Message ID: 1260**

GENERATERTKCORRECTIONS mode [port]

ASCII Example:

> generatertkcorrections rtcmv3 com2

Preconfigured set of differential corrections sent when RTCM:

> rtcm1819 ontime 1
> rtcm3 ontime 10
> rtcm22 ontime 10
> rtcm23 ontime 60
> rtcm24 ontime 60

Preconfigured set of differential corrections sent when RTCMV3:

> rtcm1004 ontime 1
> rtcm1012 ontime 1
> rtcm1006 ontime 10
> rtcm1008 ontime 10
> rtcm1033 ontime 10

Preconfigured set of differential corrections sent when RTCA:

> rtcaobs2 ontime 1
> rtcaref ontime 10

Preconfigured set of differential corrections sent When CMR:

> cmrobs ontime 1
> cmrgloobs ontime 1
> cmrref ontime 10

Preconfigured set of differential corrections sent when NOVATELX COM2:

> novatelxobs ontime 1

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	GENERATE RTKCORRE CTIONS 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	mode	RTCM	2	Serial port interface mode identifier. For more information, see *Table 37, Serial Port Interface Modes* on *page 186*	Enum	4	H
		RTCA	3				
		CMR	4				
		RTCMV3	14				
		NOVATELX	35				
3	port	See *Table 57, COM Port Identifiers* on *page 284*)		Port to configure (default = THISPORT)	Enum	4	H+4

2.4.58   GGAQUALITY
Customizes the GPGGA GPS quality indicator

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to customize the NMEA GPGGA GPS quality indicator. See also the GPGGA log on *page 474*.

Abbreviated ASCII Syntax: **Message ID: 691**

GGAQUALITY #entries [pos type1][qual1] [pos type2] [qual2]...

Input Example 1:

> ggaquality 1 waas 2

Makes the WAAS solution type show 2 as the quality indicator.

Input Example 2:

> ggaquality 2 waas 2 narrow_float 3

Makes the WAAS solution type show 2 and the NARROW_FLOAT solution type show 3, as their quality indicators.

Input Example 3:

> ggaquality 0

Sets all the quality indicators back to the default.

> ⓘ Some solution types, see *Table 84, Position or Velocity Type* on *page 407*, share a quality indicator. For example, converged PPP and NARROW_FLOAT all share an indicator of 5. This command can be used to customize an application to have unique indicators for each solution type. Sets all the quality indicators back to the default. Refer to *Table 105, GPS Quality Indicators* on *page 476*.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	GGAQUALITY 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	#entries	0-20		The number of position types that are being remapped (20 max)	Ulong	4	H+4
3	pos type1			The 1st position type that is being remapped	Enum	4	H+8

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
4	qual1	*See Table 84, Position or Velocity Type* on *page 407*		The remapped quality indicator value that will appear in the GPGGA log for the 1st position type	Ulong	4	H+12
5	pos type2	See *Table 84, Position or Velocity Type* on *page 407*		The 2nd position type that is being remapped, if applicable	Enum	4	H+16
6	qual2	See *Table 84, Position or Velocity Type* on *page 407*		The remapped quality indicator value that will appear in the GPGGA log for the 2nd position type	Ulong	4	H+20
...	Next solution type and quality indicator set, if applicable				Variable		

2.4.59 *GLIDEINITIALIZATIONPERIOD*
Configures the GLIDE initialization period

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets the initialization period for Relative PDP (GLIDE) when pseudorange measurements are used more heavily. During the initialization period, the PDP output position is not as smooth as during full GLIDE operation, but it helps to get better absolute accuracy at the start. The longer this period is the better absolute accuracy can be attained. The maximum period that can be set through GLIDEINITIALIZATIONPERIOD is 1200 seconds.

Abbreviated ASCII Syntax: **Message ID: 1760**

GLIDEINITIALIZATIONPERIOD initialization

Factory Default:

        glideinitializationperiod 300

ASCII Example:

        glideinitializationperiod 100

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	GLIDEINITIALIZATION PERIOD 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	initialization	0 -1200 s		Initialization period for GLIDE in seconds	Double	8	H

2.4.60 GLOECUTOFF
Sets GLONASS satellite elevation cut-off

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the elevation cut-off angle for tracked GLONASS satellites. The receiver does not start automatically searching for a satellite until it rises above the cut-off angle (when satellite position is known). Tracked satellites that fall below the cut-off angle are no longer tracked unless they were manually assigned (see the ASSIGN command starting on *page 72*).

In either case, satellites below the GLOECUTOFF angle are eliminated from the internal position and clock offset solution computations.

This command permits a negative cut-off angle and can be used in the following situations:

- The antenna is at a high altitude and can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction

1. Care must be taken when using GLOECUTOFF because the signals from lower elevation satellites are traveling through more atmosphere and are therefore degraded. Use of satellites below 5 degrees is not recommended.

2. Use the ELEVATIONCUTOFF command (*page 144*) to set the cut-off angle for all other systems.

3. For the OEM617D and FlexPak6D receivers, this command applies to both the primary and secondary antennas.

Abbreviated ASCII Syntax: **Message ID: 735**

GLOECUTOFF angle

Factory Default:

gloecutoff 5.0

ASCII Example:

gloecutoff 0

Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	GLO-ECUTOFF 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	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

2.4.61   HDTOUTTHRESHOLD
Controls GPHDT log output

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to control the output of the NMEA GPHDT heading log, see *page 491*. It sets a heading standard deviation threshold. Only heading information with a standard deviation less than this threshold can be output into a GPHDT message.

Abbreviated ASCII Syntax: **Message ID: 1062**

HDTOUTTHRESHOLD thresh

Factory Default:

hdtoutthreshold 2.0

ASCII Example:

hdtoutthreshold 12.0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	HDTOUT-THRESHOLD 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	thresh	0.0 - 180.0		Heading standard deviation threshold (degrees)	Float	4	H

2.4.62 HEADINGOFFSET
Adds heading and pitch offset values

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to add an offset in the heading and pitch values of the HEADING, HEADING2 and GPHDT logs. See pages *501*, *503* and *491* for more information.

Abbreviated ASCII Syntax: **Message ID: 1082**

HEADINGOFFSET headingoffsetindeg [pitchoffsetindeg]

Factory Default:

headingoffset 0 0

ASCII Example:

headingoffset 2 -1

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	HEADINGOFFSET 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	headingoffsetindeg	-180.0 - 180.0		Offset added to heading output (degrees). Default=0	Float	4	H
3	pitchoffsetindeg	-90.0 - 90.0		Offset added to pitch output (degrees). Default=0	Float	4	H+4

2.4.63   HPSEED
Specifies the initial OmniSTAR HP/XP position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This OmniSTAR HP/XP command is used to specify the initial position for OmniSTAR HP/XP. It allows you to specify the datum and undulation for the position entered. Position is then transformed into the datum currently set in the receiver. You can use STORE or RESTORE as a variable.

> The HPSEED command does not get saved when you use the SAVECONFIG command. Rather, if STORE is issued with the HPSEED command, it stores it in NVM. The RESTORE variable resends the stored HPSEED command.

Abbreviated ASCII Syntax: **Message ID: 782**

HPSEED mode [lat lon hgt latσ lonσ hgtσ [datum undulation]]

Factory Default:

hpseed reset

ASCII Examples:

To store the current HP/XP position so that it can be used as the seed in the future:

hpseed store

To use the stored HP/XP position as the seed:

hpseed restore

To use a known position in the native datum of OmniSTAR HP/XP as the seed:

hpseed set 51.11633810554 -114.03839550586 1048.2343 0.0086,0.0090,0.0191

To use a known position from a datum other than the native OmniSTAR HP/XP datum as the seed:

hpseed set 51.11633810554 -114.03839550586 1048.2343 0.0086,0.0090,0.0191 canada egm96

> 1. HP/XP seeding is restarting the HP/XP filter from known coordinates with a known accuracy as a starting point such that it is already converged. This is implemented by using the HPSEED command. There are two ways of using our implementation of HP/XP seeding:
>

a. Seed HP/XP from a stored HP/XP position:

- When HP/XP is converged and the vehicle is stopped, enter HPSEED STORE to save the current HP/XP position to NVM.
- When the vehicle is restarted, enter HPSEED RESTORE to feed the previously known position into the HP/XP process so it can start from the previous accuracy.

b. Seed HP/XP from an externally generated known position and accuracy:

- Consider the case of survey customers who enter the known antenna location with HPSEED SET <latitude> <longitude> <msl height> <lat stdev> <long stdev> <height stdev>
- If the source of the position is in a different datum than the native datum of HP/XP, or if a different undulation has been used, the transformation can be specified after <height stdev> with <datum id> <undulation type>.

Note: Initial position estimate for HP/XP and fallback when HP/XP is lost:

When HP/XP starts up, it requests the current position to get itself started. In the startup time line implemented, this is the first valid position available when the task running HP/XP receives its first L-Band data. This may or may not be a Virtual Base Station (VBS) position when VBS is also enabled. It depends on how things start up, for instance, whatever pseudorange filter position is available is used. If you want to hold off on HP/XP using the position estimate until you've confirmed that the VBS corrections have started and plenty of satellites are in the solution, start up with PSRDIFFSOURCE OMNISTAR and RTKSOURCE NONE, wait for the condition of the VBS position to be satisfactory and then set RTKSOURCE OMNISTAR as well. The HP/XP startup will be waiting until you set the RTKSOURCE. This may give some minor improvement to the convergence time of HP/XP.

This is somewhat related to the position falling back to VBS when HP/XP is lost. If both PSRDIFFSOURCE OMNISTAR and RTKSOURCE OMNISTAR are set, the BESTPOS log contains the best available of the two. There is normally an offset between the HP/XP solution and VBS.

Table 36: Seeding Mode

Binary Value	ASCII Mode Name	Description
0	RESET	Clear current seed and restart HP/XP [a]
1	SET	Specify a position and inject it into HP/XP as seed
2	STORE	Store current HP/XP position in NVM for use as a future seed [a]
3	RESTORE	Inject NVM-stored position into HP/XP as seed [a]

a. No further parameters are needed in the syntax.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	HPSEED 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	mode	See *Table 36, Page 180*		Seeding mode	Enum	4	H
3	lat	±90		Latitude (degrees)	Double	8	H+4
4	lon	±360		Longitude (degrees)	Double	8	H+12
5	hgt	-1000 to 20000000		Height above mean sea level (m)	Double	8	H+20
6	latσ			Latitude standard deviation (m)	Float	4	H+28
7	lonσ			Longitude standard deviation (m)	Float	4	H+32
8	hgtσ			Height standard deviation (m)	Float	4	H+36
9	datum	See *Table 23, Page 119*		Datum ID (default = WGS84)	Enum	4	H+40
10	undulation	see the UNDULATION command's *option* field values on *page 335*		Undulation type (default = TABLE) Refer to *Table 24, Datum Transformation Parameters* on *page 120*	Enum	4	H+44

2.4.64 *HPSTATICINIT*
Sets OmniSTAR HP/XP static initialization

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to enable or disable static initialization of OmniSTAR HP/XP. If the OmniSTAR HP/XP process knows that the receiver is stationary, it can converge more quickly.

> (i) If the HP/XP filter perceives receiver motion, it may abort static initialization. See the Static Initialization Mode bit in the HP/XP Status field of the LBANDSTAT log, details starting on *page 519*, to confirm that static initialization is in progress.

Abbreviated ASCII Syntax: **Message ID: 780**

HPSTATICINIT switch

Factory Default:

hpstaticinit disable

ASCII Example:

hpstaticinit enable

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	HPSTATICINIT 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	switch	DISABLE	0	The receiver is not stationary	Enum	4	H
		ENABLE	1	The receiver is stationary			

2.4.65 ICOMCONFIG
Configures IP virtual COM port

OEM Platform: 628, 638, FlexPak6, ProPak6

This command is used for Ethernet set up and to configure the transport/application layer of the configuration.

> (i) Access to the ICOM ports can be restricted by turning on ICOM security using the `IPSERVICE` command (see *page 192*).

Abbreviated ASCII Syntax: **Message ID: 1248**

ICOMCONFIG [port] protocol [endpoint[bindinterface]]

Factory Default:

> icomconfig icom1 tcp :3001
> icomconfig icom2 tcp :3002
> icomconfig icom3 tcp :3003

ASCII Example:

> icomconfig icom1 tcp :2000 all

> (i) Due to security concerns, the configuration ICOMCONFIG UDP :3000 and TCP should only be used by customers on a closed system, that is, board-to-board. NovAtel is not liable for any security breaches that may occur if not used on a closed system.

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	ICOMCONFIG 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	THISPORT	6	Name of the port (default = THISPORT).	Enum	4	H
		ICOM1	23				
		ICOM2	24				
		ICOM3	25				
3	protocol	DISABLED	1	Will disable the service	Enum	4	H+4
		TCP	2	Use Raw TCP			
		UDP	3	Use Raw UDP			

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
4	endpoint	Host:Port For example: 10.0.3.1:8000 mybase.com:3000		Endpoint to wait on, or to connect to where host is a host name or IP address and port is the TCP/UDP port number. If host is blank, act as a server	String [80]	variable[a]	H+8
5	bindInterface	ALL (default)	1	Not supported. Set to *ALL* for future compatibility.	Enum	4	H+88

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.66   INTERFACEMODE
Sets receive or transmit modes for ports

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used 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. For example, to accept RTCA differential corrections, set the receive type on the port to RTCA.

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, i.e., RTCA, RTCM, RTCMV3 or CMR. Disabling a port prompt is also useful when the port is connected to a modem or other device that responds with data the receiver does not recognize.

> For applications running in specific interface modes, see *Table 37* on *page 186*, please set the appropriate interface modes before sending or receiving corrections. It is important that the port interface mode matches the data being received on that port. Mismatches between the interface mode and received data can result in CPU overloads.

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. When GENERIC is set for a port, it is also disabled but 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. See *Section 3.2.89* on *page 558* for details on these logs along with the *Operation* chapter in the OEM6 Family Installation and Operation User Manual (OM-20000128) for information about pass-through logging. See also the SERIALCONFIG command on *page 282*.

> If you intend to use the SERIALCONFIG command, ensure you do so before the INTERFACEMODE command on each port. The SERIALCONFIG command can remove the INTERFACEMODE command setting if the baud rate is changed after the interface mode is set. You should also turn break detection off using the SERIALCONFIG command, see *Section 2.4.127* on *page 282*, to stop the port from resetting because it is interpreting incoming bits as a break command. If such a reset happens, the Interface mode will be set back to the default NOVATEL mode for both input and output.

OmniSTAR External Stream

This feature allows you to use OmniSTAR VBS, HP or XP when you are not tracking an L-Band signal on the OEM6. This is useful on an L-Band capable receiver where the OmniSTAR signals are unavailable. There is an OmniSTAR option for the INTERFACEMODE command (OMNISTAR), see *Table 37* on *page 186*.

For example, set the incoming INTERFACEMODE command to OMNISTAR on COM2:

```
INTERFACEMODE COM2 OMNISTAR NONE
```

where COM2 is expecting raw OmniSTAR L-Band data from an external source.

> 1. OMNISTAR is not a valid setting for an `INTERFACEMODE` output command.
>
> 2. Receiver data only comes from one port at a time.
>
> 3. When setting up a demodulator, use this command to output all raw L-Band.

Abbreviated ASCII Syntax: **Message ID: 3**

INTERFACEMODE [port] rxtype txtype [responses]

Factory Default:

interfacemode com1 novatel novatel on
interfacemode com2 novatel novatel on
interfacemode com3 novatel novatel on
interfacemode aux novatel novatel on
interfacemode usb1 novatel novatel on
interfacemode usb2 novatel novatel on
interfacemode usb3 novatel novatel on
interfacemode xcom1 novatel novatel on
interfacemode xcom2 novatel novatel on
interfacemode xcom3 novatel novatel on
interfacemode icom1 novatel novatel on
interfacemode icom2 novatel novatel on
interfacemode icom3 novatel novatel on
interfacemode ncom1 rtcmv3 none off
interfacemode ncom2 rtcmv3 none off
interfacemode ncom3 rtcmv3 none off

ASCII Example 1:

interfacemode com1 rtca novatel on

ASCII Example 2:

interfacemode com2 mrtca none

> Are NovAtel receivers compatible with others on the market?
>
> All GNSS receivers output two solutions: position and time. The manner in which they output them makes each receiver unique. Most geodetic and survey grade receivers output the position in electronic form (typically RS-232), which makes them compatible with most computers and data loggers. All NovAtel receivers have this ability. However, each manufacturer has a unique way of formatting the messages. A NovAtel receiver is not directly compatible with a Trimble or Ashtech receiver (which are also incompatible with each other) unless everyone uses a standard data format.
>
> But there are several standard data formats available. For position and navigation output there is the NMEA format. Real-time differential corrections use RTCM or RTCA format. For receiver code and phase data RINEX format is often used. NovAtel and all other major manufacturers support these formats and can work together using them. The NovAtel format measurement logs can be converted to RINEX using the utilities provided in NovAtel Connect.

Table 37: Serial Port Interface Modes

Binary Value	ASCII Mode Name	Description
0	NONE	The port accepts/generates nothing. The port is disabled
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	OMNISTAR	The port accepts OMNISTAR corrections, see also *OmniSTAR External Stream* on *page 185*
6-7	Reserved	
8	RTCMNOCR	RTCM with no CR/LF appended [a]
9	Reserved	
10	TCOM1	INTERFACEMODE tunnel modes. To configure a full duplex tunnel, configure the baud rate on each port. Once a tunnel is established, the baud rate does not change. Special characters, such as a BREAK condition, do not route across the tunnel transparently and the serial port is altered, see the `SERIALCONFIG` command on *page 282*. Only serial ports may be in a tunnel configuration: COM1, COM2, COM3 or AUX may be used. For example, configure a tunnel at 115200 bps between COM1 and AUX: COM AUX 115200 COM COM1 115200 INTERFACEMODE AUX TCOM1 NONE OFF INTERFACEMODE COM1 TAUX NONE OFF The tunnel is fully configured to receive/transmit at a baud rate of 115200 bps
11	TCOM2	
12	TCOM3	
13	TAUX [b]	

Binary Value	ASCII Mode Name	Description
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
16-17	Reserved	
18	GENERIC	The port accepts/generates nothing. SEND/SENDHEX commands from another port generate data on this port. Any incoming data on this port can be seen with PASSCOM logs on another port, see *page 558*
19	Reserved	
20	MRTCA	The port accepts/generates Modified Radio Technical Commission for Aeronautics (MRTCA) corrections
21-26	Reserved	
27	AUTO	For auto-detecting different RTK correction formats
35	NOVATELX	The port accepts/generates NOVATELX corrections

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.

b. Only available on specific models.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	INTERFACEMODE 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	See *Table 28, Communications Port Identifiers* on *page 141*		Serial port identifier (default = THISPORT)	Enum	4	H
3	rxtype	See *Table 37* on *page 186*		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)			

2.4.67　IOCONFIG
Sets the behavior of multiplexed I/O pins

OEM Platform: 638, ProPak6

The IOCONFIG command controls the settings of the multiplexed pins on the OEM638 headers. This command provides independent control over each multiplexed pin.

Abbreviated ASCII Syntax:　　　　　　　**Message ID: 1663**

IOCONFIG pmux_pin switch function

ASCII Example:

ioconfig pvout enable

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	IOCONFIG 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	pmux_pin	PVOUT	0	Selects the multiplexed pin to change. See *Table 38, PMUX Pin Description* on *page 189* for more information. USERIO2 is not available on the ProPak6	Enum	4	H
		ERRORFLAG	1				
		NRSETOUT	2				
		USERIO2	3				
3	switch	Disable	0	Disables MUX control on the pin entered in PMUX_pin. See *Table 38, PMUX Pin Description* on *page 189* for more information.	Enum	4	H+4
		Enable	1	Enables MUX control on the pin entered in PMUX_pin. See *Table 38, PMUX Pin Description* on *page 189* for more information.			
4	function	IN	0	This field is used if *PMUX_pin* is set to USERIO2. See *Table 38, PMUX Pin Description* on *page 189* for more information. This field is not used if PMUX_Pin is set to PVOUT, ERRORFLAG or NRSETOUT.	Enum	4	H+8
		OUT	1				
		EVENT	2				

Table 38: PMUX Pin Description

ASCII	Binary	Switch Value	Description
PVOUT	0	Enable (Default)	Position Valid Output. Pin 14 on the OEM638 expansion header presents the Position Valid indicator. On the ProPak6, I?O port pin 5 presents the Position Valid indicator.
		Disable	OEM6 expansion header Pin 14 or ProPak6 I/O port pin 5 is the EVENT_OUT3 output.
ERRORFLAG	1	Enable (Default)	Error Flag Output. Pin 38 on the OEM638 main header or ProPak6 I/O pin 8 is set high when any error is detected as described in RXSTATUS (See RXSTATUS log on *page 659*).
		Disable	OEM638 main header Pin 38 or ProPak6 I/O pin 8 is the EVENT_IN4 input.
NRSETOUT	2	Enable (Default)	An active low reset pulse is output on pin 28 of the OEM638 expansion header or ProPak6 I/O pin 7 when the board resets.
		Disable	Pin 28 on the OEM638 expansion header is the EVENT_IN2 input; or ProPak6 I/O pin 7 is the EVENT_IN3 input.
USERIO2	3	Enable (Default)	If the Function field is set to 0 "IN", the USERIO2 pin (Main header pin 31) is the GPIO input and the expansion header pin 28 is either EVENT_IN2 or NRESETOUT, depending on the last NRESETOUT MUX selection. **Note**: The last NRESETOUT MUX selection will be RESETOUT if the last USERIO2 selection was EVENT. **Note**: USERIO2 is not accessible on the Pro
			If the Function field is set to 1 "OUT", the USERIO2 pin is the GPIO output and the expansion header pin 28 is either EVENT_IN2 or NRESETOUT, depending on the last NRESETOUT MUX selection. **Note**: The last NRESETOUT MUX selection will be RESETOUT if the last USERIO2 selection was EVENT.
			If the Function field is set to 2 "EVENT", the USERIO2 pin is the EVENT_IN2 and the expansion header pin 28 is always NRESETOUT.
		Disable	The USERIO2 pin is a GPIO in the last selected direction and expansion header pin 28 is EVENT_IN2. **Note**: The last selected direction will be IN if the last USERIO2 MUX selection was EVENT.

2.4.68 IONOCONDITION
Sets ionospheric condition

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to change the level of ionosphere activity that is assumed by the RTK positioning algorithms.

> Only advanced users should use this command.

Abbreviated ASCII Syntax: **Message ID: 1215**

IONOCONDITION mode

Factory Default:

ionocondition auto

ASCII Example:

ionocondition normal

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	IONOCONDITION header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively		H	
2	mode	QUIET	0	Receiver assumes a low level of ionosphere activity	Enum	4	H
		NORMAL	1	Receiver assumes a medium level of ionosphere activity			
		DISTURBED	2	Receiver assumes a high level of ionosphere activity			
		AUTO	10	Receiver monitors the ionosphere activity and adapts behavior accordingly			

2.4.69 IPCONFIG
Configures network IP settings

OEM Platform: 628, 638, FlexPak6, ProPak6

This command is used to configure static/dynamic TCP/IP properties for the Ethernet connection.

> (i) In addition to configuring an IP address and netmask for the interface, this command also includes a gateway address. However, the receiver will only use the gateway address if the Ethernet interface is configured as the preferred network interface (see SETPREFERREDNETIF command on *page 299*).

Abbreviated ASCII Syntax: **Message ID: 1243**

IPCONFIG [interface_name] address_mode [ip_address [netmask [gateway]]]

Factory Default:

ipconfig etha dhcp

ASCII Examples:

ipconfig etha static 192.168.74.10 255.255.255.0 192.168.74.1

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	IPCONFIG 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	interface name	ETHA	2	Name of the Ethernet interface (default = ETHA)	Enum	4	H
3	address mode	DHCP	1	Use Dynamic IP address	Enum	4	H+4
		STATIC	2	Use Static IP address			
4	IP address	ddd.ddd.ddd.ddd (For example: 10.0.0.2)		IP Address-decimal dot notation	String [16]	variable[a]	H+8
5	netmask	ddd.ddd.ddd.ddd (For example: 255.255.255.0)		Netmask-decimal dot notation	String [16]	variable[a]	H+24
6	gateway	ddd.ddd.ddd.ddd (For example: 10.0.0.1)		Gateway-decimal dot notation	String [16]	variable[a]	H+40

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.70 IPSERVICE
Configure availability of networks ports/services

OEM Platform: 628, 638, ProPak6

Use the `IPSERVICE` command to configure the availability of specific network ports/ services. When disabled, the service does not accept incoming connections.

> ⓘ By default, the FTP Server is disabled.

Abbreviated ASCII Syntax: **Message ID: 1575**

IPSERVICE ipservice switch

Factory Detault:

ipservice disabled

ASCII Example:

ipservice ftp_server enable

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	IPSERVICE 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	ipservice	NO_PORT	0	No port	Enum	4	H
		FTP_SERVER	1	FTP server port. (Default = disabled) **Not applicable to the OEM628 platform**			
		WEB_SERVER	2	Web server port (Default=disabled) **Not applicable to the OEM628 platform**			
		SECURE_ICOM	3	Enables or disables security on ICOM ports. When security is enabled, a login is required as part of the connection process (see LOGIN command on *page 209*). *Note*: Security in this sense means users must supply a name and password before being allowed to enter commands on the ICOM ports. It does not mean there is data encryption			
3	switch	DISABLE	0	Disable the IP service specified.	Enum	4	H+4
		ENABLE	1	Enable the IP service specified.			

2.4.71   LEDCONFIG
LED Controller Configuration Interface

OEM Platform: ProPak6

Use the `LEDCONFIG` command to configure some of the ProPak6 front panel LED indicators.

Five of the LEDs on the Propak6 front panel have configurable threshold settings that determine when the LED changes color states. The five configurable LEDs are SATTRACK1, SATTRACK2, GNSS1, GNSS2 and DATALOG.

The configuration settings for the SATTRACK1, SATTRACK2 and DATALOG LEDs take two parameters: one value determines when the LED turns red and the other value determines when it turns amber (with the default setting outside the value range being green).

Some ProPak6 models contain two GNSS receiver cards to calculate alignment and heading. GNSS1 refers to the primary receiver card in the ProPak6 and GNSS2 refers to the optional secondary receiver card in the ProPak6. The GNSS1 and GNSS2 LEDs only take one parameter. There are several subcategories for each that can be set to specific LED states (color and blink or solid display). There are 5 Position Types which categorize the full range of receiver position types. When the first parameter passed to the LEDCONFIG command is either GNSS1 or GNSS2, you are setting all five position types to the same LED pattern. If you want to specify an LED pattern for any of the five GNSS sub categories, which correspond to the five position types, use the GNSSx_ parameter (for example GNSS1_SINGLE). See the examples on the following page.

Abbreviated ASCII Syntax: **Message ID: 1498**

LEDCONFIG ledid [param 1] [param 2]

ASCII Example:

ledconfig datalog 20 40

Field#	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	LEDCONFIG header	Command Header			0
2	LEDID	*Table 39, LED ID* on *page 195*	Enum	4	H
3	Param 1	Parameter value. Exact meaning depends on LED ID.	Long	4	H+4
4	Param 2	Parameter value. Exact meaning depends on LED ID.	Long	4	H+8

(i)   The `LOG LEDCONFIG` command displays the current configuration of all LEDs.

Examples:

        LEDCONFIG DATALOG 20 40

This command configures the LED as follows:

- • red: available memory <20%
- • amber: available memory ≥20% and ≤40%
- • green: available memory >40%

        LEDCONFIG SATTRACK1 3 5

This command configures the LED as follow:

- • red: 3 or fewer satellites
- • amber: 4 or 5 satellites
- • green: 6 or more satellites

To restore the GNSS1 LED to its default configuration for all position states:

        LEDCONFIG GNSS1 0

To configure the GNSS2 LED so *No Fix* is OFF, *Single Point* is AMBER, and everything else is GREEN, enter the following commands:

        LEDCONFIG GNSS2 3
        LEDCONFIG GNSS2_NOFIX 1
        LEDCONFIG GNSS2_SINGLE 5

An alternate way to configure the GNSS2 LED so *No Fix* is OFF, *Single Point* is AMBER, and everything else is GREEN, enter the following commands:

        LEDCONFIG GNSS2_NOFIX 1
        LEDCONFIG GNSS2_SINGLE 5
        LEDCONFIG GNSS2_CONVERGING 3
        LEDCONFIG GNSS2_CONVERGED 3
        LEDCONFIG GNSS2_PSRPDP 3

To set just the GNSS2 Converging Accuracy to its default pattern:

```
LEDCONFIG GNSS2_CONVERGING 0
```

Table 39: LED ID

Binary	ASCII	Description	Param 1	Param 2
1	SATTRACK1	Set thresholds for SV Tracking LED for card 1	Red LED SV threshold When the number of satellites tracked by card 1 is less than or equal to this value, the LED is red.	Amber LED SV threshold When the number of satellites tracked by card 1 is less than or equal ot this value, the LED is amber.
2	SATTRACK2	Set thresholds for SV Tracking LED for card 2	Red LED SV threshold When the number of satellites tracked by card 2 is less than or equal to this value, the LED is red.	Amber LED SV threshold When the number of satellites tracked by card 2 is less than or equal to this value, the LED is amber.
3	GNSS1	Assign LED pattern for all Position Status states for card 1	LED pattern (see *Table 40* on *page 196*)	N/A
4	GNSS2	Assign LED pattern for all Position Status states for card 2	LED pattern (see *Table 40* on *page 196*)	N/A
9	DATALOG	Set thresholds for Logging Capacity LED	Red LED free space % threshold. When the percentage of free memory space is below this value the LED is red.	Amber LED free space % threshold When the percentage of free memory space is less than or equal to this value the LED is amber.
19	GNSS1_NOFIX	Assign the LED pattern for the *No Fix* position state for card 1	LED pattern (see *Table 40* on *page 196*)	N/A
20	GNSS1_SINGLE	Assign the LED pattern for the *Single Point* position state for card 1	LED pattern (see *Table 40* on *page 196*)	N/A
21	GNSS1_CONVERGING	Assign the LED pattern for the *Converging Accuracy* position state for card 1	LED pattern (see *Table 40* on *page 196*)	N/A
22	GNSS1_CONVERGED	Assign the LED pattern for the *Converged Accuracy* position state for card 1	LED pattern (see *Table 40* on *page 196*)	N/A
23	GNSS1_PSRPDP	Assign the LED pattern for the *PSR/PDP Using Corrections* position state for card 1	LED pattern (see *Table 40* on *page 196*)	N/A

Binary	ASCII	Description	Param 1	Param 2
24	GNSS2_NOFIX	Assign the LED pattern for the *No Fix* position state for card 2	LED pattern (see *Table 40* on *page 196*)	N/A
25	GNSS2_SINGLE	Assign the LED pattern for the *Single Point* position state for card 2	LED pattern (see *Table 40* on *page 196*)	N/A
26	GNSS2_CONVERGING	Assign the LED pattern for the *Converging Accuracy* position state for card 2	LED pattern (see *Table 40* on *page 196*)	N/A
27	GNSS2_CONVERGED	Assign the LED pattern for the *Converged Accuracy* position state for card 2	LED pattern (see *Table 40* on *page 196*)	N/A
28	GNSS2_PSRPDP	Assign the LED pattern for the *PSR/PDP Using Corrections* position state for card 2	LED pattern (see *Table 40* on *page 196*)	N/A

Table 40: GNSS1/GNSS2 LED Patterns

Position Status	Value
Default value	0
Off	1
Flashing Green	2
Solid Green	3
Flashing Amber	4
Solid Amber	5

Table 41: SATTRACK1/SATRACK2 LED default values

SATTRACK Threshold	LED State
Tracking 0 SVs	Off
Tracking 1 to 3 SVs	Red
Tracking 4 to 5 SVs	Amber
Tracking 6 SVs or more	Green

Table 42: DATALOG LED default values

DATALOG Threshold	LED State
Available internal memory capacity is greater than 40%	Green
Available internal memory capacity is between 20% and 40%	Amber
Available internal memory capacity is less than 20%	Red
System Busy (format, chkdsk, etc)	Alternating Green and Amber

Table 43: GNSS1/GNSS2 LED default values

GNSS State	LED State
No Fix	Off
Single Point	Flashing Amber
Converging Accuracy	Amber
Converged Accuracy	Green
PSR/PDP Using Corrections	Flashing Green

2.4.72 LOCALIZEDCORRECTIONDATUM
Sets a local datum

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to select a localized correction datum before using localized wide area corrections. The choices are World Geodetic System 84 (WGS84) and North American 1983 (NAD83) including Areas 37-42. The default is WGS84, however, when the receiver receives OmniSTAR data and a LOCALIZEDCORRECTIONDATUM NAD83 command is issued, it bases the localized wide area corrections on NAD83

> (i) RTCM corrections are always with respect to the datum selected at the base. For example, if the LOCALIZEDCORRECTIONDATUM is set to NAD83 at a base station, the datum of the positions produced at the rover receiver using these localized corrections will be NAD83. This is true even though the datum in the rover BESTPOS log shows WGS84.

Localized Wide Area Corrections Mode

The local wide area corrections enhancement allows a NovAtel receiver to receive OmniSTAR VBS corrections, compute an equivalent DGPS correction and then output it in RTCM format to any GNSS receiver. The corrections can be selected to be output in either in the WGS84 or NAD83 datum.

Localized OmniSTAR corrections are available on products with L-Band capability. Supported datums provide these corrections with WGS84 as the default.

This enhancement also uses the following logs:

RTCMOMNI1, see *page 636* and *OmniSTAR Local Wide Area Corrections on page 639*

Use the SAVECONFIG command to save local wide area corrections interface settings.

Abbreviated ASCII Syntax: **Message ID: 947**

LOCALIZEDCORRECTIONDATUM type

ASCII Example:

localizedcorrectiondatum nad83

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	LOCALIZED-CORRECTION-DATUM 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

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	type	WGS84	1	Localized correction datum type	Enum	4	H
		NAD83	2				

2.4.73 LOCKOUT
Prevents the receiver from using a satellite

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to prevent the receiver from using a satellite in the solution computations.

> (i)
> 1. The LOCKOUT command does not prevent the receiver from tracking an undesirable satellite.
> 2. LOCKOUT and UNLOCKOUT commands can be used with GPS, GLON-ASS, SBAS and QZSS PRNs.

This command must be repeated for each satellite to be locked out. See also the UNLOCKOUT (*page 337*) and UNLOCKOUTALL (*page 338*) commands.

Abbreviated ASCII Syntax: **Message ID: 137**

LOCKOUT prn

Input Example:

> lockout 8

> (i)
> The LOCKOUT command removes one or more satellites from the solution while leaving other satellites available.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	LOCKOUT 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	prn	GPS: 1-32 SBAS: 120-138, 183-187 GLONASS: see *1.3 GLONASS Slot and Frequency Numbers* on *page 34.* QZSS: 193-197		Unique identifier for the satellite being locked out	Ulong	4	H

2.4.74 LOCKOUTSYSTEM
Prevents the receiver from using a system

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to prevent the receiver from using satellites in a system in the solution computations.

> ⓘ The LOCKOUTSYSTEM command does not prevent the receiver from tracking an undesirable satellite.

This command must be repeated for each system to be locked out. See also the UNLOCKOUTSYSTEM on *page 339* and UNLOCKOUTALL on *page 338*.

Abbreviated ASCII Syntax: **Message ID: 871**

LOCKOUTSYSTEM system

Factory Defaults:

  lockoutsystem galileo
  lockoutsystem sbas
  lockoutsystem qzss

> ⓘ The LOCKOUTSYSTEM command removes one or more systems from the solution while leaving other systems available.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	LOCKOUT SYSTEM 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	system	See *Table 110, Satellite System* on *page 508*	A single satellite system to be locked out		Enum	4	H

> ⓘ Refer to *Table 110, Satellite System* on *page 508* for a list satellite systems.

2.4.75   LOG
Requests logs from the receiver

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Many different types of data can be logged using different methods of triggering the log events. Every log element can be directed to any combination of the receiver's ports. The ONTIME trigger option requires the addition of the *period* parameter. See Chapter 3, Data Logs on page *358* for further information and a complete list of data log structures. The LOG command tables in this section show the binary format followed by the ASCII command format.

The optional parameter [hold] prevents a log from being removed when the `unlogall` command, with its defaults, is issued. To remove a log which was invoked using the [hold] parameter requires the specific use of the `UNLOG` command, see *page 340*. To remove all logs that have the [hold] parameter, use the `UNLOGALL` command with the *held* field set to 1, see *page 342*.

The [port] parameter is optional. If [port] is not specified, [port] is defaulted to the port that the command was received on.

> (i)   1.   The OEM6 family of receivers can handle 64 simultaneous log requests. If an attempt is made to log more than 64 logs at a time, the receiver responds with an Insufficient Resources error.
>
> 2.   The user is cautioned that each log requested requires additional CPU time and memory buffer space. Too many logs may result in lost data and low CPU idle time. Receiver overload can be monitored using the idle-time field and buffer overload bits of the Receiver Status in any log header.
>
> 3.   Only the MARKPOS, MARK2POS, MARKTIME or MARK2TIME 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.
>
> 4.   Use the ONNEW trigger with the MARKTIME, MARK2TIME, MARKPOS or MARK2POS logs.
>
> 5.   Polled log types do not allow fractional offsets or ONTIME rates faster than 1 Hz.
>
> 6.   If ONTIME trigger is used with asynchronous logs, the time stamp in the log does not necessarily represent the time the data was generated but rather the time when the log is transmitted.

Abbreviated ASCII Syntax: **Message ID: 1**

LOG [port] message [trigger [period [offset [hold]]]]

Factory Default:

> log com1 rxstatuseventa onnew 0 0 hold
> log com2 rxstatuseventa onnew 0 0 hold
> log com3 rxstatuseventa onnew 0 0 hold
> log aux rxstatuseventa onnew 0 0 hold
> log usb1 rxstatuseventa onnew 0 0 hold
> log usb2 rxstatuseventa onnew 0 0 hold
> log usb3 rxstatuseventa onnew 0 0 hold
> log icom1 rxstatuseventa onnew 0 0 hold
> log icom2 rxstatuseventa onnew 0 0 hold
> log icom3 rxstatuseventa onnew 0 0 hold

Abbreviated ASCII Example 1:

> log com1 bestpos ontime 7 0.5 hold

The above example shows BESTPOS logging to COM port 1 at 7 second intervals and offset by 0.5 seconds (output at 0.5, 7.5, 14.5 seconds and so on). The [hold] parameter is set so that logging is not disrupted by the UNLOGALL command (*page 342*).

To send a log once, the trigger option can be omitted.

Abbreviated ASCII Example 2:

> log com1 bestpos once 0.000000 0.000000 nohold

See *Section 2.1 Command Formats* on *page 40* for additional examples.

> ⓘ Using the NovAtel Connect utility there are two ways to initiate data logging from the receiver's serial ports. Either enter the LOG command in the *Console* window or use the interface provided in the *Logging Control* window. Ensure the Power Settings on the computer are not set to go into Hibernate or Standby modes. Data is lost if one of these modes occurs during a logging session.

Field	Field Type	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	LOG (binary) header	See *Table 3, Binary Message Header Structure* on *page 26*	This field contains the message header	-	H	0
2	port	See *Table 4*	Output port	Enum	4	H
3	message	Any valid message ID	Message ID of log to output (refer to *Table 81, OEM6 Logs in Numerical Order* on *page 375*) for a list of message ID numbers	Ushort	2	H+4
4	message type	Bits 0-4 = Reserved Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response Bit (*page 33*) 0 = Original Message 1 = Response Message	Message type of log	Char	1	H+6
5	Reserved			Char	1	H+7
6	trigger	0 = ONNEW	Does not output current message but outputs when the message is updated (not necessarily changed)	Enum	4	H+8
		1 = ONCHANGED	Outputs the current message and then continues to output when the message is changed			
		2 = ONTIME	Output on a time interval			
		3 = ONNEXT	Output only the next message			
		4 = ONCE	Output only the current message			
		5 = ONMARK	Output when a pulse is detected on the mark 1 input, MK1I [a] [b]			
7	period	Valid values for the high rate logging are 0.05, 0.1, 0.2, 0.25 and 0.5. For logging slower than 1 Hz any integer value is accepted	Log period (for ONTIME trigger) in seconds [c]	Double	8	H+12

Field	Field Type	Binary Value	Description	Format	Binary Bytes	Binary Offset
8	offset	Offset for period (ONTIME trigger) in seconds. To log data at 1 second, after every minute, set the period to 60 and the offset to 1	A valid value is any integer (whole number) smaller than the period. These decimal values, on their own, are also valid: 0.1, 0.2, 0.25 or 0.5, as well as any multiple of the maximum logging rate defined by the receiver model. Values less than 1ms will be considered an offset of 0 ms. The offset cannot be smaller than the minimum measurement period supported by the model.	Double	8	H+20
9	hold	0 = NOHOLD	Allow log to be removed by the UNLOGALL command	Enum	4	H+28
		1 = HOLD	Prevent log from being removed by the default UNLOGALL command			

a. Refer to the Technical Specifications appendix in the OEM6 Family Installation and Operation User Manual (OM-20000128) for more details on the MK1I pin. ONMARK only applies to MK1I. Events on MK2I (if available) do not trigger logs when ONMARK is used. Use the ONNEW trigger with the MARKTIME, MARK2TIME, MARKPOS or MARK2POS logs.

b. Once the 1PPS signal has hit a rising edge, for both MARKPOS and MARKTIME logs, a resolution of both measurements is 10 ns. As for the ONMARK trigger for other logs that measure latency, for example RANGE and POSITION logs such as BESTPOS, it takes typically 20-30 ms (50 ms maximum) for the logs to output information from the 1PPS signal. Latency is the time between the reception of the 1PPS pulse and the first byte of the associated log. See also the MARKPOS and MARKTIME logs starting on *page 533*.

c. See Appendix A in the OEM6 Family Installation and Operation User Manual (OM-20000128) for the maximum raw measurement rate to calculate the minimum period. If the value entered is lower than the minimum measurement period, the command will be rejected.

Field	Field Name	ASCII Value	Description	Field Type
1	LOG (ASCII) header	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII or ASCII respectively	-
2	port	*Table 4, Detailed Port Identifier* on *page 27*	Output port (default = THISPORT)	Enum

Field	Field Name	ASCII Value	Description	Field Type
3	message	Any valid message name, with an optional A or B suffix (refer to *Table 81, OEM6 Logs in Numerical Order* on *page 375*)	Message name of log to output	Char [ ]
4	trigger	ONNEW	Output when the message is updated (not necessarily changed)	Enum
		ONCHANGED	Output when the message is changed	
		ONTIME	Output on a time interval	
		ONNEXT	Output only the next message	
		ONCE	Output only the current message (default)	
		ONMARK	Output when a pulse is detected on the mark 1 input, MK1I [a][b]	
5	period	Any positive double value larger than the receiver's minimum raw measurement period	Log period (for ONTIME trigger) in seconds (default = 0) (see *Footnote c* on *page 205*)	Double
6	offset	Any positive double value smaller than the period	Offset for period (ONTIME trigger) in seconds. If you want to log data, at 1 second after every minute, set the period to 60 and the offset to 1 (default = 0)	Double
7	hold	NOHOLD	To be removed by the UNLOGALL command (default)	Enum
		HOLD	Prevent log from being removed by the default UNLOGALL command	

a. Refer to the Technical Specifications appendix in the OEM6 Family Installation and Operation User Manual (OM-20000128) for more details on the MK1I pin. ONMARK only applies to MK1I. Events on MK2I (if available) do not trigger logs when ONMARK is used. Use the ONNEW trigger with the MARKTIME, MARK2TIME, MARKPOS or MARK2POS logs.

b. Once the 1PPS signal has hit a rising edge, for both MARKPOS and MARKTIME logs, a resolution of both measurements is 10 ns. As for the ONMARK trigger for other logs that measure latency, for example RANGE and POSITION logs such as BESTPOS, it takes typically 20-30 ms (50 ms maximum) for the logs to output information from the 1PPS signal. Latency is the time between the reception of the 1PPS pulse and the first byte of the associated log. See also the MARKPOS and MARKTIME logs starting on *page 533*.

2.4.76   LOGFILE
Opens/closes log files in internal flash

OEM Platform: 638, ProPak6

Use the LOGFILE command to open and close a log file, recorded on internal flash storage. To record logs, log requests are made to be redirected to the FILE port.

If no file name is supplied, one is automatically generated based on the following format:

<PSN>_index.LOG
where
PSN is the PSN of the receiver and
index is an incrementing number starting at 1

The first unused number starting from 1 is selected on subsequent commands (for example, if there are 5 automatically named log files (names ending in 1-5) and log file 3 is deleted, the next automatic file name ends in 3 since it is not in use).

Example: NOV12001200A_2.LOG
In this example "NOV12001200A" is the receiver PSN and "2" indicates that this is the second auto-named file in the system.

- A new COMPORT_ENUM called FILE represents the internal logging to flash memory. It has a NOVATEL Interface Mode output only, no input is possible.
- Once a log file is opened, any logs requested for the FILE port are recorded to the data log file on internal flash memory.
- Only logs published after the log file is open are recorded.
- Only one log file can be open at a time.
- Logs requested to the FILE port are produced even if the log file is closed (they do not go anywhere nor are recorded). If a new log file is opened, recording of the previously requested logs continues with the new file.
- The DOSCMD command is not permitted when a log file is open for writing.
- The logging button on a Propak6 simply issues the LOGFILE OPEN/CLOSE command (toggling the current state) and auto-generates a file name based on the description above.
- The LOGFILESTATUS log (*page 525*) reports the current state of the logging system.

Abbreviated ASCII Syntax: **Message ID: 157**

LOGFILE pdc_fileactionenum [filename]

ASCII Example:

logfile open   - creates an auto-named file for recording data

logfile open flightpath.dat   - creates a file named flightpath.dat for recording data

logfile close   - closes the currently open log file

log file rawimusb onnew   - records rawimusb logs to internal flash if a file has been opened for writing

Field	Field Type	Binary Value	ASCII Value	Description	Format	Binary Bytes	Binary Offset
1	LOGFILE 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	PDC_FileActionEnum	0	OPEN	Open (create) new logging file	Enum	4	H
		1	CLOSE	Close logging file			
3	Filename			File name (optional)	Char	128	H+4

2.4.77 LOGIN
Start a secure ICOM connection to the receiver

OEM Platform: 628, 638, FlexPak6, ProPak6

When ICOM ports have security enabled (see the `IPSERVICE` command on *page 192*), a session to the ICOM port can be established but commands are refused until a valid `LOGIN` command is issued. Both the UserName and Password are required. The `LOGIN` command checks the supplied credentials against known UserNames/Passwords and determines if the login is successful or not. A successful login permits the secured ICOM command interpreter to accept further commands and returns OK. An unsuccessful login does not release the secured ICOM command interpreter and returns Login Failed.

Entering a `LOGIN` command on any command port other than the ICOM port has no effect, regardless of whether the UserName/Password is correct. In this case, the appropriate response (OK or Login Failed) is returned, but there is no effect on the command interpreter.

> (i) When security is enabled, access to the port is restricted unless a valid name and password are supplied. It does not mean there is data encryption enabled. Username is case-insensitive and password is case-sensitive.

Abbreviated ASCII Syntax: **Message ID: 1671**

LOGIN [commport] username password

ASCII Example:

      login admin adminpassword

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	LOGIN 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	commport	ICOM1	23	The ICOM port to log into. This is an optional parameter. If no value is entered, logs in to the ICOM port currently being used. (default=THISPORT)	Enum	4	H
		ICOM2	24				
		ICOM3	25				
3	username			Provide the user name for the login command. The user name is not case sensitive.	String [32]	variable[a]	H + 4
4	password			Provide the password for the user name. The password is case sensitive	String [28]	variable[a]	variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.78 LOGOUT
End a secure ICOM session started using the LOGIN command

OEM Platform: 628, 638, FlexPak6, ProPak6

Use the LOGOUT command to sign out of an ICOM connection after a user has successfully logged in using the LOGIN command. After the sending the LOGOUT command, the ICOM connection will not accept further commands, other than a new LOGIN command. The session itself is not ended. This only applies to ICOM ports that have had security enabled (see the IPSERVICE command on *page 192*).

Abbreviated ASCII Syntax: **Message ID: 1672**

LOGOUT [commport]

ASCII Example:

> logout

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	LOGOUT 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	commport	ICOM1	23	The ICOM port from which to log out. This is an optional parameter. If no value is entered, logs out from the ICOM port currently being used.	Enum	4	H
		ICOM2	24				
		ICOM3	25				

2.4.79 MAGVAR
Sets a magnetic variation correction

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The receiver computes directions referenced to True North (also known as geodetic north). The Magnetic Variation Correction command (MAGVAR) is used to navigate in agreement with magnetic compass bearings. The correction value entered here causes the "bearing" field of the NAVIGATE log to report bearing in degrees Magnetic. The receiver computes the magnetic variation correction when using the auto option. See *Figure 5, Illustration of Magnetic Variation and Correction on Page 212.*

The receiver calculates values of magnetic variation for given values of latitude, longitude and time using the International Geomagnetic Reference Field (IGRF) 2010 spherical harmonic coefficients and IGRF time corrections to the harmonic coefficients. The model is intended for use up to the year 2015. The receiver will compute for years beyond 2015 but accuracy may be reduced.

Abbreviated ASCII Syntax: **Message ID: 180**

MAGVAR type correction [std dev]]

Factory Default:
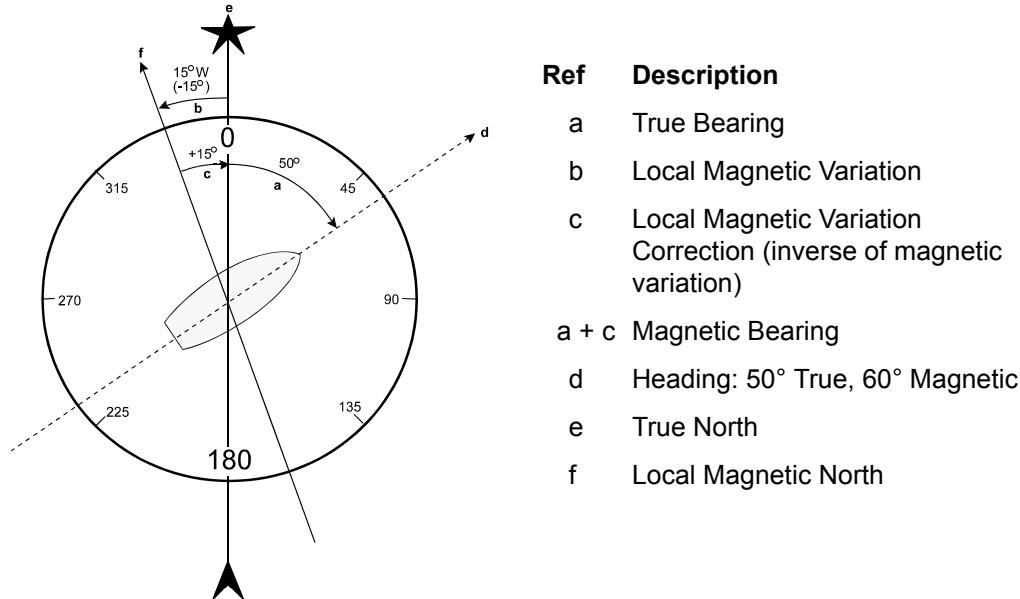
        magvar correction 0 0

ASCII Example 1:

        magvar auto

ASCII Example 2:

        magvar correction 15 0

Figure 5: Illustration of Magnetic Variation and Correction

Ref	Description
a	True Bearing
b	Local Magnetic Variation
c	Local Magnetic Variation Correction (inverse of magnetic variation)
a + c	Magnetic Bearing
d	Heading: 50° True, 60° Magnetic
e	True North
f	Local Magnetic North

How does GNSS determine what Magnetic North is? Do the satellites transmit a database or some kind of look up chart to determine the declination for your given latitude and longitude? How accurate is it?

Magnetic North refers to the location of the Earth's Magnetic North Pole. Its position is constantly changing in various cycles over centuries, years and days. These rates of change vary and are not well understood. However, we are able to monitor the changes.

True North refers to the earth's spin axis, that is, at 90° north latitude or the location where the lines of longitude converge. The position of the spin axis does not vary with respect to the Earth.

The locations of these two poles do not coincide. Thus, a relationship is required between these two values for users to relate GNSS bearings to their compass bearings. This value is called the magnetic variation correction or declination.

GNSS does not determine where Magnetic North is nor do the satellites provide magnetic correction or declination values. However, OEM6 receivers store this information internally in look up tables so that when you specify that you want to navigate with respect to Magnetic North, this internal information is used. These values are also available from various information sources such as the United States Geological Survey (USGS). The USGS produces maps and has software which enables the determination of these correction values. By identifying your location (latitude and longitude), you can obtain the correction value. Refer to our book An Introduction to GNSS, available on our website at www.novatel.com/support/ for USGS contact information.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	MAGVAR 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	type	AUTO	0	Use IGRF corrections	Enum	4	H
		CORRECTION	1	Use the correction supplied			
3	correction	± 180.0 degrees		Magnitude of correction (Required field if type = Correction)	Float	4	H+4
4	std_dev	± 180.0 degrees		Standard deviation of correction (default = 0)	Float	4	H+8

2.4.80   MARKCONTROL
Controls processing of mark inputs

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to control the processing of the mark inputs. Using this command, the mark inputs can be enabled or disabled, polarity can be changed and a time offset and guard against extraneous pulses can be added.

The MARKPOS and MARKTIME logs have their outputs (and extrapolated time tags) pushed into the future (relative to the mark input (MKI) event) by the amount entered into the time bias field. In almost all cases, this value is set to 0, which is also the default setting (*page 533* and *page 536*).

Abbreviated ASCII Syntax: **Message ID: 614**

MARKCONTROL signal [switch [polarity [timebias [timeguard]]]]
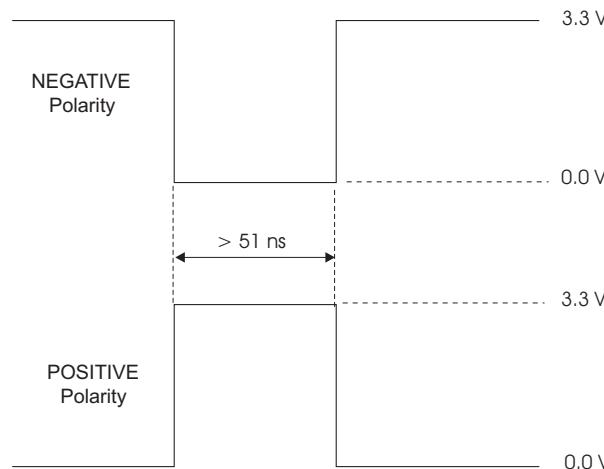
Factory Default:

> markcontrol mark1 enable
> markcontrol mark2 enable

ASCII Example:

> markcontrol mark1 enable negative 50 100

Figure 6: TTL Pulse Polarity

> If using an external device, such as a camera, connect the device to the receiver's I/O port. Use a cable that is compatible to both the receiver and the device. A MARKIN pulse can be a trigger from the device to the receiver. See also the MARKPOS and MARKTIME logs (*page 533* and *page 536*).

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	MARKCONTROL 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	signal	MARK1	0	Specifies which mark input the command should be applied to. Set to MARK1 for the MK1I input, MARK2 for MK2I, MARK3 for MK3I and MARK4 for MK4I. All of the mark inputs have 10 K pull-up resistors to 3.3 V and are leading edge triggered	Enum	4	H
		MARK2	1				
		MARK3 (OEM638 & ProPak6 only)	2				
		MARK4 (OEM638 & ProPak6 only)	3				
3	switch	DISABLE	0	Disables or enables processing of the mark input signal for the input specified. If DISABLE is selected, the mark input signal is ignored (default = ENABLE)	Enum	4	H+4
		ENABLE	1				
4	polarity	NEGATIVE	0	Optional field to specify the polarity of the pulse to be received on the mark input. See Figure 6, TTL Pulse Polarity on Page 214 for more information (default= NEGATIVE)	Enum	4	H+8
		POSITIVE	1				
5	timebias	Any valid long value		Optional value to specify an offset, in nanoseconds, to be applied to the time the mark input pulse occurs (default =0)	Long	4	H+12
6	timeguard	default: 4 minimum: 2 Any valid ulong value larger than the receiver's minimum raw measurement period [a]		Optional field to specify a time period, in milliseconds, during which subsequent pulses after an initial pulse are ignored	Ulong	4	H+16

a. See *Appendix A* in the [OEM6 Family Installation and Operation User Manual](link) (OM-20000128) for the maximum raw measurement rate to determine the minimum period. If the value entered is lower than the minimum measurement period, the value is ignored and the minimum period is used.

2.4.81   MODEL
Switches to a previously authorized model

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to switch the receiver between models previously added with the AUTH command (*page 82*). When the MODEL command is issued, the receiver saves the specified model as the active model. The active model is now used on every subsequent start up. The MODEL command causes an automatic reset.

Use the VALIDMODELS log to output a list of available models on the receiver. The VALIDMODELS log is described on *page 738*. Use the VERSION log to output the active model, see *page 742.*

> (i)   If you switch to an expired model, the receiver will reset and enter into an error state. You will need to switch to a valid model to continue.

Abbreviated ASCII Syntax: **Message ID: 22**

MODEL model

Input Example:

model d2lr0rccr

> (i)   NovAtel uses the term models to refer to and control different levels of functionality in the receiver firmware. For example, a receiver may be purchased with an L1 only capability and be easily upgraded at a later time to a more feature intensive model, like L1/L2 dual-frequency. All that is required to upgrade is an authorization code for the higher model and the AUTH command (see *page 82*). Reloading the firmware or returning the receiver for service to upgrade the model is not required. Upgrades are available from NovAtel Customer Support.

.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	MODEL 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	model	Max 16 character null-terminated string (including the null)		Model name	String [max 16]	Variable[a]	Variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.82   MOVINGBASESTATION
Enables the use of a moving base station

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to enable or disable a receiver from transmitting corrections without a fixed position.

The moving base function allows you to obtain a centimeter level xyz baseline estimate when the base station and possibly the rover are moving. It is very similar to normal RTK, with one stationary base station and a moving rover (refer to *Transmitting and Receiving Corrections* section of the *Operation* chapter in the [OEM6 Family Installation and Operation User Manual](#) (OM-20000128)). The BSLNXYZ log is an asynchronous 'matched' log that can be logged with the onchanged trigger to provide an accurate baseline between the base and rover.

Due to the latency of the reference station position messages, the following logs are not recommended to be used when in moving baseline mode: BESTXYZ, GPGST, MARKPOS, MARK2POS, MATCHEDPOS, MATCHEDEYZ, RTKPOS and RTKXYZ. The position error of these logs could exceed 100 m, depending on the latency of the reference station position message. If a rover position is required during moving basestation mode, then PSRPOS is recommended.

The MOVINGBASESTATION command must be used to allow the base to transmit messages without a fixed position.

ⓘ 1. Use the PSRPOS position log at the rover. It provides the best accuracy and standard deviations when the MOVINGBASESTATION mode is enabled.

2. This command supports RTCM V2.3 messages (except RTCM2021), RTCM V3 operation and CMR GLONASS.

3. RTCA, RTCM1819 and RTCM V3 support includes GPS + GLONASS operation.

4. The MOVINGBASESTATION mode is functional if any of the following RTK message formats are in use: RTCAOBS, RTCAOBS2, CMROBS, RTCAREF or CMRREF.

Abbreviated ASCII Syntax: **Message ID: 763**

MOVINGBASESTATION switch

Factory Default:

movingbasestation disable
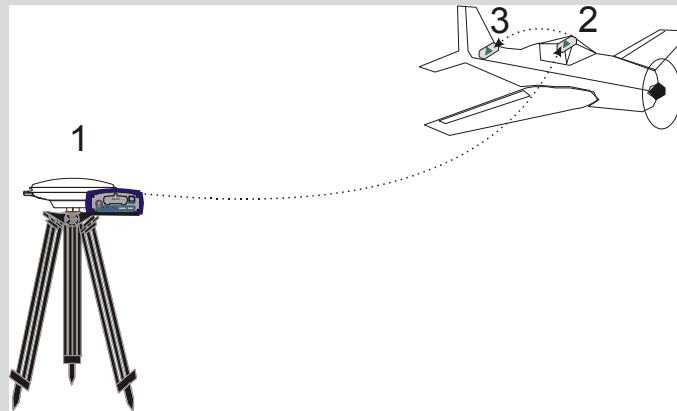
ASCII Example:

movingbasestation enable

1. Consider the case where there is a fixed base, an airplane flying with a moving base station near its front and a rover station at its tail end.
Corrections can be sent between the receivers in a 'daisy chain' effect, where the fixed base station sends corrections to the moving base station, which in turn can send corrections to the rover.

Figure 7: Moving Base Station 'Daisy Chain' Effect

> When using this method, the position type is only checked at the fixed base station. Moving base stations will continue to operate under any conditions.

2. This command is useful for moving base stations doing RTK positioning at sea. A rover station is used to map out local areas (for marking shipping lanes, hydrographic surveying and so on), while the base station resides on the control ship. The control ship may not move much (parked at sea), but there is a certain amount of movement due to the fact that it is floating in the ocean. By using the MOVING-BASESTATION command, the control ship is able to use RT2-level RTK positioning and move to new survey sites.

.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	MOVING-BASESTATION 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	switch	DISABLE	0	Do not transmit corrections without a fixed position	Enum	4	H
		ENABLE	1	Transmit corrections without a fixed position			

2.4.83   NMEATALKER
Sets the NMEA talker ID

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used 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, GPVTG and GPZDA log outputs.

The default GPS NMEA messages (nmeatalker gp) include specific information about only the GPS satellites that have a 'GP' talker solution, even when GLONASS satellites are present. The NMEATALKER AUTO command changes this behavior so that the NMEA messages include all satellites in the solution and the talker ID changes according to those satellites.

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 and the second sentence on 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.

Abbreviated ASCII Syntax: **Message ID: 861**

NMEATALKER id

Factory Default:

          nmeatalker gp

ASCII Example:

          nmeatalker auto

> ⓘ The NMEATALKER command only affects NMEA logs that are capable of a GPS output. For example, GLMLA is a GLONASS-only log and the output will always use the GL talker. *Table 44, NMEA Talkers* on *page 220* shows the NMEA logs and whether they use GPS (GP), GLONASS (GL), Galileo (GA) or combined (GN) talkers with NMEATALKER AUTO.

Table 44: NMEA Talkers

Log	Talker IDs
GLMLA	GL
GPALM	GP
GPGGA	GP
GPGLL	GP or GL or GA or GN
GPGRS	GP or GL or GA or GN
GPGSA	GP or GL or GA or GN
GPGST	GP or GL or GA or GN
GPGSV	GP and GL and GA
GPRMB	GP or GL or GA or GN
GPRMC	GP or GL or GA or GN
GPVTG	GP or GL or GA or GN
GPZDA	GP

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	NMEA-TALKER 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	ID	GP	0	GPS only	Enum	4	H
		AUTO	1	GPS, GLONASS, combined			

2.4.84 NMEAVERSION
Sets the NMEA Version for Output

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use this command to set the output version of NMEA messages.

Abbreviated ASCII Syntax: **Message ID 1574**

NMEAVERSION version

Factory Defaults:

nmeaversion v31

ASCII Example:

nmeaversion v41

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	NMEAVERSION 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	Version	V31	0	NMEA messages will be output in NMEA version 3.10 format.	Enum	4	H
		V41	1	NMEA messages will be output in NMEA version 4.10 format.			

2.4.85   NTRIPCONFIG
Configures NTRIP

OEM Platform: 628, 638, FlexPak6, ProPak6

This command sets up and configures NTRIP communication.

Abbreviated ASCII Syntax: **Message ID 1249**

NTRIPCONFIG port type [protocol [endpoint [mountpoint [username [password [bindinterface]]]]]]

> ⓘ   Mountpoint, username and password are all set up on the caster.

Factory Default:

> ntripconfig ncom1 disabled
> ntripconfig ncom2 disabled
> ntripconfig ncom3 disabled
> ntripconfig ncomx disabled

ASCII Example:

> ntripconfig ncom1 client v1 :2000 calg0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	NTRIPCONFIG 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	THISPORT	6	Name of the port see *Table 28, Communications Port Identifiers* on *page 141* (THISPORT = default)	Enum	4	H
		NCOM1	26				
		NCOM2	27				
		NCOM3	28				
3	type	DISABLED	1	NTRIP type (default DISABLED)	Enum	4	H+4
		CLIENT	2				
		SERVER	3				
4	protocol	V1	1	Protocol (default V1)	Enum	4	H+8
		V2	2				

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
5	endpoint	Max 80 character string		Endpoint to wait on or to connect to where host is a hostname or IP address and port is the TCP/UDP port number (default = 80)	String [80]	variable[a]	H+12
6	mountpoint	Max 80 character string		Which mount point to use	String [80]	variable[a]	variable
7	user name	Max 30 character string		Login user name	String [30]	variable[a]	variable
8	password	Max 30 character string		Password	String [30]	variable[a]	variable
9	bindInterface	ALL (default)	1	Not supported. Set to *ALL* for future compatibility.	Enum	4	variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.86 *NTRIPSOURCETABLE*
Set NTRIPCASTER ENDPONTS

OEM Platform: 628, 638, FlexPak6, ProPak6

This command is used to set the NTRIPCASTER ENDPOINTS to be used for the SOURCETABLE log.

Abbreviated ASCII Syntax: **Message ID 1343**

NTRIPSOURCETABLE endpoint [reserved1] [reserved2]

Factory Default:

ntripsourcetable none

ASCII Example:

ntripsourcetable hera.novatel.com:2101
ntripsourcetable 198.161.64.11:2101

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	NTRIPSOURCE TABLE 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	Endpoint	Max 80 character string		Endpoint, in format of host:port, to connect to where the host is a hostname or IP address and port is the TCP/IP port number	String [80]	variable[a]	H
3	Reserved1	Reserved		Reserved	Ulong	4	variable
4	Reserved2	Reserved		Reserved	Ulong	4	variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.87   NVMRESTORE
Restores NVM data after an NVM failure

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to restore Non-Volatile Memory (NVM) data after a NVM Fail error. This failure is indicated by bit 13 of the receiver error word being set (see also *RXSTATUS page 659* and *RXSTATUSEVENT page 668*). If corrupt NVM data is detected, the receiver remains in the error state and continues to flash an error code on the Status LED until the NVMRESTORE command is issued (refer to the chapter on *Built-In Status Tests* in the OEM6 Family Installation and Operation User Manual (OM-20000128) for further explanation).

If you have more than one auth code and the saved model is lost, then the model may need to be entered using the MODEL command or it is automatically saved in NVM on the next start up. If the almanac was lost, a new almanac is automatically saved when the next complete almanac is received (after approximately 15 minutes of continuous tracking). If the user configuration was lost, it has to be reentered by the user. This could include communication port settings.

> (i)   The factory default for the COM ports is 9600, n, 8, 1.

After entering the NVMRESTORE command and resetting the receiver, the communications link may have to be reestablished at a different baud rate from the previous connection.

Abbreviated ASCII Syntax: **Message ID: 197**

NVMRESTORE

> (i)   The possibility of NVM failure is extremely remote, however, if it should occur it is likely that only a small part of the data is corrupt. This command is used to remove the corrupt data and restore the receiver to an operational state. The data lost could be the user configuration, almanac, model or other reserved information.

2.4.88 OMNIUSEGLONASS
Enables/disables GLONASS in OmniSTAR

OEM Platform: 628, 638, FlexPak6, ProPak6

This command is used to enable or disable the use of GLONASS in OmniSTAR.

Abbreviated ASCII Syntax: **Message ID: 1199**

OMNIUSEGLONASS switch

Factory Default:

omniuseglonass enable

ASCII Example:

omniuseglonass disable

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	OMNIUSEGLONASS 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	switch	DISABLE	0	Enables/disables GLONASS in OmniSTAR	Enum	4	H
		ENABLE	1				

2.4.89 PDPFILTER
Enables, disables or resets the PDP filter

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to enable, disable or reset the Pseudorange/Delta-Phase (PDP) filter. The main advantages of the PDP implementation are:

- Smooths a jumpy position
- Bridges outages in satellite coverage (the solution is degraded from normal but there is at least a reasonable solution without gaps)

> (i) 1. Enable the PDP filter to output the PDP solution in BESTPOS, BESTVEL and NMEA logs.
>
> 2. Refer to the *Operation* chapter of the OEM6 Installation and Operation Manual (OM-20000128) for a section on configuring your receiver for PDP or GLIDE™ operation.

GLIDE Position Filter

GLIDE is a mode of the PDP[1] filter that optimizes the position for consistency over time rather than absolute accuracy. This is ideally in clear sky conditions where the user needs a tight, smooth and consistent output. The GLIDE filter works best with SBAS. The PDP filter is smoother than a least squares solution but is still noisy in places. The GLIDE filter produces a very smooth solution with relative rather than absolute position accuracy. There should typically be less than 1 centimeter difference in error from epoch to epoch. GLIDE also works in single point, DGPS and OmniSTAR VBS modes. See also the PDPMODE command on *page 229* and the PDPPOS, PDPVEL and PDPXYZ logs starting on *page 564*.

Abbreviated ASCII Syntax: **Message ID: 424**

PDPFILTER switch

Factory Default:

pdpfilter disable

ASCII Example:

pdpfilter enable

---

1. Refer also to our application note APN=038 on Pseudorange/Delta-Phase (PDP), available on our website a www.novatel.com/support/search.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	PDPFILTER 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	switch	DISABLE	0	Enable/disable/reset the PDP filter. A reset clears the filter memory so that the PDP filter can start over	Enum	4	H
		ENABLE	1				
		RESET	2				

2.4.90 PDPMODE
Selects the PDP mode and dynamics

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to select the mode and dynamics of the PDP filter.

> (i) 1. The `PDPFILTER ENABLE` command (*page 227*) must be entered before the `PDPMODE` command.
>
> 2. It is recommended that the ionotype be left at AUTO when using either normal mode PDP or GLIDE. See also `SETIONOTYPE` command on *page 295*.

Abbreviated ASCII Syntax: **Message ID: 970**

PDPMODE mode dynamics

Factory Default:

        pdpmode normal auto

ASCII Example:

        pdpmode relative dynamic

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	PDPMODE 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	mode	NORMAL	0	In relative mode, GLIDE performance is optimized to obtain a consistent error in latitude and longitude over time periods of 15 minutes or less, rather than to obtain the smallest absolute position error. See also *GLIDE Position Filter* on *page 227* for GLIDE mode additional information	Enum	4	H
		RELATIVE	1				
		GLIDE	3				
3	dynamics	AUTO	0	Auto detect dynamics mode	Enum	4	H+4
		STATIC	1	Static mode			
		DYNAMIC	2	Dynamic mode			

2.4.91 PDPVELOCITYOUT
Set the type of velocity used in the PDPVEL log

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command configures the type of velocity that is output in the PDPVEL log. By default, the PDPVELOCITYOUT mode is set to PDP while the PDPVEL log (*page 567*) and associated BESTVEL log (*page 417*) contain the velocity from the PDP filter. When the PDPVELOCITYOUT mode is set to PSR, a Doppler-based velocity (similar to that output with the PSR position) with lower latency is output.

Abbreviated ASCII Syntax: **Message ID: 1324**

pdpvelocityout mode

Factory Default:

pdpvelocityout pdp

ASCII Example:

pdpvelocityout psr

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	PDPVELOCITYOUT 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	mode	PDP	0	Use the velocity from the PDP filter.	Enum	4	H
		PSR	1	Use a Doppler-based velocity with lower latency.			

2.4.92 POSAVE
Implements base station position averaging

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command implements position averaging for base stations. Position averaging continues for a specified number of hours or until the estimated averaged position error is within specified accuracy limits. Averaging stops when the time limit or the horizontal standard deviation limit or the vertical standard deviation limit is achieved. When averaging is complete, the FIX POSITION command is automatically invoked.

If you initiate differential logging, then issue the POSAVE command followed by the SAVECONFIG command, the receiver averages positions after every power on or reset. It then invokes the FIX POSITION command to enable it to send differential corrections.

Abbreviated ASCII Syntax: **Message ID: 173**

POSAVE [state] [maxtime [maxhstd [maxvstd]]]

Factory Default:

posave off

ASCII Example 1:

posave on 24 1 2

ASCII Example 2:

posave off

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	POSAVE 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	state	ON	1	Enable or disable position averaging (default = ON)	Enum	4	H
		OFF	0				
3	maxtime	0.01 - 100 hours		Maximum amount of time that positions are to be averaged (default=0.0)	Float	4	H+4
4	maxhstd	0 - 100 m		Desired horizontal standard deviation (default = 0.0)	Float	4	H+8
5	maxvstd	0 - 100 m		Desired vertical standard deviation (default = 0.0)	Float	4	H+12

ⓘ The POSAVE command can be used to establish a new base station, in any form of survey or RTK data collection, by occupying a site and averaging the position until either a certain amount of time has passed or position accuracy has reached a user specified level. User specified requirements can be based on time or horizontal or vertical quality of precision.

2.4.93 POSTIMEOUT
Sets the position time out

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This commands is used to set the time out value for the position calculation in seconds.

In position logs, for example BESTPOS or PSRPOS, when the position time out expires, the Position Type field is set to NONE. Other field values in these logs remain populated with the last available position data. Also, the position is no longer used in conjunction with the almanac to determine what satellites are visible.

Abbreviated ASCII Syntax: **Message ID: 612**

POSTIMEOUT sec

Factory Default:

postimeout 600

ASCII Example:

postimeout 1200

> (i) When performing data collection in a highly dynamic environment (for example, urban canyons or in high speed operations), you can use POSTIMEOUT to prevent the receiver from outputting calculated positions that are too old. Use POSTIMEOUT to force the receiver position type to NONE. This ensures that the position information being used in BESTPOS or PSRPOS logs is based on a recent calculation. All position calculations are then recalculated using the most recent satellite information.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	POSTIMEOUT 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	sec	0-86400		Time out in seconds	Ulong	4	H

2.4.94 *PPPCONVERGEDCRITERIA*
Configures decision for PPP convergence

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command controls how the PPP filter determines if the solution has converged.

> ⓘ Relaxing the convergence threshold shortens the time before a PPP solution is reported as converged. However, it does not alter solution behavior. During the initial PPP solution period, the positions can have decimeter error variation. Only relax the convergence threshold if the application can tolerate higher solution variability.

Abbreviated ASCII Syntax: **Message ID: 1566**

PPPCONVERGEDCRITERIA criteria tolerance

Factory Default:

pppconvergedcriteria horizontal_stddev 0.20

ASCII Example:

pppconvergedcriteria total_stddev 0.15

Table 45: Convergence Criteria

ASCII	Binary	Description
TOTAL_STDDEV	1	Use the total, 3D, standard deviation
HORIZONTAL_STDDEV	2	Use the horizontal, 2D, standard deviation

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	PPP CONVERGED CRITERIA header	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see *Section 1.1, Message Types on Page 21*)	-	H	0
2	Criteria	Criteria (see Table 45 above)	Enum	4	H
3	Tolerance	Tolerance (m)	Float	4	H+4

2.4.95 PPPDYNAMICS
Sets the PPP dynamics mode

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command configures the dynamics assumed by the PPP filter. AUTO detects the antenna dynamics and adapts filter operation accordingly.

> ⓘ The automatic dynamics detection, however, may be fooled by very slow, "creeping" motion, where the antenna consistently moves less than 2 cm/s. In such cases, the mode should explicitly be set to DYNAMIC.

Abbreviated ASCII Syntax: **Message ID: 1551**

PPPDYNAMICS mode

Factory Default:

     pppdynamics dynamic

ASCII Example:

     pppdynamics auto

Table 46: Dynamics Mode

ASCII	Binary	Description
AUTO	0	Automatically determines dynamics mode
STATIC	1	Static mode
DYNAMIC	2	Dynamic mode

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	PPPDYNAMICS header	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see Section 1.1, Message Types on Page 21)	-	H	0
2	Mode	Dynamics mode (refer to *Table 46, Dynamics Mode* on *page 234*)	Enum	4	H

2.4.96 PPPSEED
Control the seeding of the PPP filter

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The `PPPSEED` command controls the seeding of the PPP filter. Accurate position seeding can accelerate PPP convergence.

PPPSEED SET is used to explicitly specify a seed position. The seed position must be in a datum consistent with the PPP corrections that will be used. For NovAtel CORRECT with PPP, this is ITRF2008.

> ⓘ Caution must be exercised when using PPPSEED SET. While a good seed position can accelerate convergence, a bad seed position hurts performance. In some cases, a bad seed can prevent a solution from ever converging to a correct position. In other cases, a bad seed might be rejected immediately. In still other cases, the filter might operate with it for a time period only to reject it later. In this case, the filter position is partially reset, with a corresponding discontinuity in the PPP position.

PPPSEED STORE and RESTORE are intended to simplify seeding in operations where the antenna does not move between power-down and power-up. For example, in agricultural operations a tractor might be stopped in a field at the end of a day and then re-started the next day in the same position. Before the receiver is powered-down, the current PPP position could be saved to NVM using the `PPPSEED STORE` command, and then that position applied as a seed after power-up using PPPSEED RESTORE.

PPPSEED AUTO automates the STORE and RESTORE process. When this option is used, the PPP filter automatically starts using the stopping position of the previous day. For this command to work, the `PPPDYNAMICS` command (*page 234*) setting must be AUTO so that the receiver can determine when it is static, or the filter must explicitly be told it is static using PPPDYNAMIC STATIC. Additionally, in order for the receiver to recall the saved seed, the PPPSEED AUTO command should be saved to NVM using the `SAVECONFIG` command (*page 271*).

Abbreviated ASCII Syntax: **Message ID: 1544**

PPPSEED option [latitude longitude height northing_std._dev. easting_std._dev. height_std._dev.]

ASCII Example:

        pppseed set 51.11635322441 -114.03819311672 1064.5458 0.05 0.05 0.05

Table 47: PPP Seeding Option

ASCII	Binary	Description
CLEAR	0	Resets the stored seed, and prevents any auto seeding from occurring.
SET	1	Immediately apply the specified co-ordinates as a seed position.
STORE	2	Store the current PPP position in NVM for use as a future seed.
RESTORE	3	Retrieve and apply a seed position that was previously saved in NVM via the STORE or AUTO options.
AUTO	4	Automatically store and restore PPP seed positions.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	PPPSEED header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see Section 1.1, Message Types on Page 21)	-	H	0
2	option	see Table 47 above		Seeding option	Enum	4	H
3	latitude	±90		Latitude (degrees)	Double	8	H+4
4	longitude	±180		Longitude (degrees)	Double	8	H+12
5	height	> -2000.0		Ellipsoidal height (metres)	Double	8	H+20
6	northing std. dev.			Northing standard deviation (metres)	Float	4	H+28
7	easting std. dev.			Easting standard deviation (metres)	Float	4	H+32
8	height std. dev.			Ellipsoidal height standard deviation (metres)	Float	4	H+36

2.4.97 *PPPSOURCE*
Specifies the PPP correction source

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, FlexPak6D, ProPak6

This command determines what corrections the PPP filter will use. When transitioning between explicitly specified sources, there can be some delay between this command being accepted and the source specified in the PPP solution changing.

▼ The AUTO source behavior is subject to change.

Abbreviated ASCII Syntax: **Message ID: 1707**

PPPSOURCE source

Factory Default:

pppsource auto

ASCII Example:

pppsource none

Table 48: PPP Source Types

ASCII	Binary	Description
NONE	0	Disable the PPP filter
TERRASTAR	1	Only accept TerraStar PPP corrections
VERIPOS	2	Only accept Veripos PPP corrections
PACE	3	Only accept NovAtel PACE PPP corrections
TERRASTAR_D	6	Only accept TerraStar-D PPP corrections
AUTO	100	Automatically select and use the best corrections

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	PPPSOURCE header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see Section 1.1, Message Types on Page 21)	-	H	0
2	source	see Table 48 above		Source option	Enum	4	H

2.4.98 PPPTIMEOUT
Sets the maximum age of the PPP corrections

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets the maximum age of the corrections used in the PPP filter. Corrections older than the specified duration are not applied to the receiver observations and uncorrected observations are not used in the filter.

Abbreviated ASCII Syntax: **Message ID: 1560**

PPPTIMEOUT delay

Factory Default:

      ppptimeout 360

ASCII Example:

      ppptimeout 120

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	PPPTIMEOUT header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see Section 1.1, Message Types on Page 21)	-	H	0
2	delay	5 to 900 s		Maximum corrections age	ULong	4	H

2.4.99 PPSCONTROL
Controls the PPS output

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command provides a method for controlling the polarity, period and pulse width of the PPS output on the OEM6. The PPS output can also be disabled using this command.

The leading edge of the 1 PPS pulse is always the trigger/reference:

> PPSCONTROL ENABLE NEGATIVE

generates a normally high, active low pulse with the falling edge as the reference, while:

> PPSCONTROL ENABLE POSITIVE

generates a normally low, active high pulse with the rising edge as the reference.

The pulse width is user-adjustable. The adjustable pulse width feature generates these uses for the PPS signal:

- Supporting triggers/systems that need longer, or shorter, pulse widths than the default to register the pulse enabling a type of GPIO line for manipulation of external hardware control lines

Abbreviated ASCII Syntax: **Message ID: 613**

PPSCONTROL [switch [polarity [period [pulse width]]]]

Factory Default:

> ppscontrol enable negative 1.0 1000

ASCII Example:

> ppscontrol enable positive 0.5 2000

> ⓘ This command is used to setup the PPS signal coming from the receiver. For example, to take measurements such as temperature or pressure, in synch with your GNSS data, the PPS signal can be used to trigger measurements in other devices.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	PPSCONTROL 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	switch	DISABLE	0	Disable the PPS	Enum	4	H+4
		ENABLE	1	Enable the PPS (default)			
		ENABLE_FINE TIME	2	Enable the PPS only when FINE or FINESTEERING time status has been reached			
		ENABLE_FINE TIME_MINUTE ALIGN	3	Enable the PPS only when FINE or FINESTEERING time status has been reached AND the start of the next 60 seconds (1 minute modulus) has occurred			
3	polarity	NEGATIVE	0	Optional field to specify the polarity of the pulse to be generated on the PPS output. See *Figure 6* on *Page 214* for more information (default= NEGATIVE)	Enum	4	H+8
		POSITIVE	1				
4	period	0.05, 0.1, 0.2, 0.25, 0.5, 1.0, 2.0, 3.0,...20.0		Optional field to specify the period of the pulse, in seconds (default=1.0)	Double	8	H+12
5	pulse width	Any positive value less than or equal to half the period		Optional field to specify the pulse width of the PPS signal in microseconds. This value should always be less than or equal to half the period (default=1000)	Ulong	4	H+20

2.4.100 PPSCONTROL2

Controls polarity, period, pulse width and estimated error limit of the PPS output

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The PPSControl2 command provides a method for controlling the polarity, period, pulse width, and estimated error limit of the PPS output on the OEM6. The PPS output can also be disabled using this command.

This command is identical to the PPSCONTROL command (*page 239*) with the addition of two new switch settings and a new parameter that represents the Estimated Error Limit.

The two new switch states allow more control over disabling/enabling the PPS. The ENABLE_FINETIME switch prevents the PPS from being enabled until FINE or FINESTEERING time status has been reached. The ENABLE_FINETIME_MINUTEALIGN switch is similar to ENABLE_FINETIME with caveat that the PPS will still not be enabled until the start of the next 60 seconds (a 1 minute modulus) after FINE or FINESTEERING time status has been reached.

The estimated error limit sets an allowable ± range for the clock offset. The PPS output is only enabled when the clock offset is within this range.

Abbreviated ASCII Syntax: **Message ID: 1740**

ppscontrol2 [switch [polarity [period [pulse width [estimated error limit]]]]]

Factory default:

       ppscontrol2 enable negative 1.0 1000 0

ASCII Example:

       ppscontrol2 enable_finetime positive 0.5 2000 10

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	PPSCONTROL2 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

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	switch	DISABLE	0	Disable the PPS	Enum	4	H+4
		ENABLE	1	Enable the PPS (default)			
		ENABLE_FINETIME	2	Enable the PPS only when FINE or FINESTEERING time status has been reached			
		ENABLE_FINETIME _MINUTEALIGN	3	Enable the PPS only when FINE or FINESTEERING time status has been reached AND the start of the next 60 seconds (1 minute modulus) has occurred			
3	polarity	NEGATIVE	0	Optional field to specify the polarity of the pulse to be generated on the PPS output. See *Figure 6* on *Page 214* for more information (default = NEGATIVE).	Enum	4	H+8
		POSITIVE	1				
4	period	0.05, 0.1, 0.2, 0.25, 0.5, 1.0, 2.0, 3.0,...20.0		Optional field to specify the period of the pulse in seconds (default = 1.0).	Double	8	H+12
5	pulse width	Any value less than or equal to half the pulse period in microseconds.		Optional field to specify the pulse width of the PPS signal in microseconds. This value should always be equal to half the period (default = 1000).	Ulong	4	H+20

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
6	estimated error limit	0 to 2147483647 in nanoseconds		Optional field to specify the ± estimated error limit (in nanoseconds) for the clock offset (default = 0). The PPS output is only enabled when the clock offset is within this limit. An estimated error limit of 0 removes the estimated error limit restraint on the PPS.	Long	4	H+24

2.4.101 PROFILE
Profile in Non-Volatile Memory (NVM)

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to configure multiple profiles in the NVM at receiver startup. The output is in the PROFILEINFO log (*page 576*). See also the FRESET command (*page 164*).

Abbreviated ASCII Syntax: **Message ID: 1411**

PROFILE option name [command]

ASCII Examples:

```
PROFILE create Base
PROFILE createelement Base "log versiona"
PROFILE createelement Base "serialconfig com2 115200"
PROFILE createelement Base "log com2 rtca1 ontime 1"
PROFILE activate Base
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	PROFILE 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	Option	refer to *Table 49, Profile Option* on *page 245*		Profile options	Enum	4	H
3	Name			Profile name	String [Max 20]	variable[a]	H+4
4	Command			Profile command	String [Max 200]	variable[a]	variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

> Restrictions:
> 1. Only supports up to a maximum of 9 profiles.
>
> 2. Only supports up to a maximum of 20 commands per profile.
>
> 3. Only supports up to a maximum of 200 characters long for each command.
>
> 4. Only supports up to a maximum of 1500 characters for all commands in one profile.
>
> 5. If one of the profiles is activated, the SAVECONFIG functionality is disabled.
>
> 6. All profile data cleared from the NVM only by using the `FRESET` or `NVMC-LEAR` commands.
>
> 7. The receiver resets after a profile is activated.

Table 49: Profile Option

Binary	ASCII	Description
0	Reserved	
1	CREATE	Create a profile
2	DELETE	Delete an existing profile
3	CREATEELEMENT	Create an element in an existing profile
4	DELETEELEMENT	Delete an existing element in an existing profile
5	ACTIVATE	Activate an existing profile
6	DEACTIVATE	Deactivate a running profile

2.4.102 PSRDIFFSOURCE
Sets the pseudorange differential correction source

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to identify which base station to accept differential corrections from. This is useful when the receiver is receiving corrections from multiple base stations. See also the RTKSOURCE command on *page 266*.

> (i)
> 1. When a valid PSRDIFFSOURCE command is received, the current correction is removed immediately rather than in the time specified in PSRDIFFSOURCETIMEOUT (*page 249*).
>
> 2. To use L-Band differential corrections, an L-Band receiver and the OmniS-TAR VBS service or use of a DGPS service is required. Contact NovAtel for details.
>
> 3. For ALIGN users: the ALIGN rover will not use RTK corrections automatically to do PSRDIFF positioning, as ALIGN is commonly used with a moving base. If you have a static base and want a PSRDIFF position, at the ALIGN rover, set the PSRDIFFSOURCE RTK.

Abbreviated ASCII Syntax: **Message ID: 493**

PSRDIFFSOURCE type [id]

Factory Default:

      psrdiffsource auto any

ASCII Examples:

    1. Enable only SBAS:
       rtksource none
       psrdiffsource sbas
       sbascontrol enable auto

    2. Enable OmniSTAR VBS and HP or XP:
       rtksource omnistar
       psrdiffsource omnistar

    3. Enable RTK and PSRDIFF from RTCM, with a fall-back to SBAS:
       rtksource rtcm any
       psrdiffsource rtcm any
       sbascontrol enable auto

    4. Disable all corrections:
       rtksource none
       psrdiffsource none

> ⓘ Since several errors affecting signal transmission are nearly the same for two receivers near each other on the ground, a base at a known location can monitor the errors and generate corrections for the rover to use. This method is called Differential GPS and is used by surveyors to obtain submetre accuracy.
>
> Major factors degrading GPS signals, which can be removed or reduced with differential methods, are atmospheric, satellite orbit errors and satellite clock errors. Errors not removed include receiver noise and multipath.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	PSRDIFFSOURCE 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	type	See *Table 50, DGPS Type* on *page 248*	ID Type. All types (except NONE) may revert to SBAS (if enabled) or SINGLE position types. See *Table 84, Page 407* [a]	Enum	4	H
3	Base station ID	Char [5] or ANY	ID string	Char[5]	8 [b]	H+4

a. If ANY is chosen, the receiver ignores the ID string. Specify a Type when using base station IDs.

b. In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.

Table 50: DGPS Type

Binary	ASCII	Description
0	RTCM [d]	RTCM ID: 0 ≤ RTCM ID ≤ 1023 **or** ANY
1	RTCA [d]	RTCA ID: A four character string containing only alpha (a-z) or numeric characters (0-9) **or** ANY
2	CMR [d][a]	CMR ID: 0 ≤ CMR ID ≤ 31 **or** ANY
3	OMNISTAR [b][d]	In the PSRDIFFSOURCE command, OMNISTAR enables OmniSTAR VBS and disables other DGPS types. OmniSTAR VBS produces RTCM-type corrections. In the RTKSOURCE command, OMNISTAR enables OmniSTAR HP/XP (if allowed) and disables other RTK types. OmniSTAR HP/XP has its own filter, which computes corrections to within about 10 cm accuracy
Reserved		
5	SBAS [c][d]	In the PSRDIFFSOURCE command, when enabled, SBAS such as WAAS, EGNOS and MSAS, forces the use of SBAS as the pseudorange differential source. SBAS is able to simultaneously track two SBAS satellites and incorporate the SBAS corrections into the position to generate differential quality position solutions. An SBAS-capable receiver permits anyone within the area of coverage to take advantage of its benefits. **If SBAS is set in the RTKSOURCE command, it can not provide carrier phase positioning and returns an error**
6	RTK [c]	In the PSRDIFFSOURCE command, RTK enables using RTK correction types for PSRDIFF positioning. The correction type used is determined by the setting of the RTKSOURCE command
10	AUTO [c][d]	In the PSRDIFFSOURCE command, AUTO means that if any correction format is received then it will be used. If multiple correction formats are available, then RTCM, RTCA and RTK will be preferred over OmniSTAR, which will be preferred over SBAS messages. If RTCM, RTCA and RTK are all available then the type of the first received message will be used. In the RTKSOURCE command, AUTO means that both the NovAtel RTK filter and the OmniSTAR HP/XP filter (if authorized) are enabled. The NovAtel RTK filter selects the first received RTCM, RTCA, RTCMV3 or CMR message. The BESTPOS log selects the best solution between NovAtel RTK and OmniSTAR HP/XP
11	NONE [c][d]	Disables all differential correction types
12	Reserved	
13	RTCMV3 [b][a]	RTCM Version 3.0 ID: 0 ≤ RTCMV3 ID ≤ 4095 **or** ANY
14	NOVATELX	NovAtel proprietary message format ID: A four character string containing alpha (a-z) or numeric characters (0-9) or ANY

  a. This cannot be used in the `PSRDIFFSOURCE` command.

  b. Base station ID parameter is ignored.

  c. Available only with the `PSRDIFFSOURCE` command.

  d. All PSRDIFFSOURCE entries fall back to SBAS (except NONE).

2.4.103 PSRDIFFSOURCETIMEOUT
Sets pseudorange differential correction source timeout

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

When multiple differential correction sources are available, this command allows the user to set a time in seconds, that the receiver will wait before switching to another differential source, if corrections from the original source are lost.

Abbreviated ASCII Syntax: **Message ID: 1449**

PSRDIFFSOURCETIMEOUT option [timeout]

Factory Default:

       psrdiffsourcetimeout auto

ASCII Example:

       psrdiffsourcetimeout auto

       psrdiffsourcetimeout set 180

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	PSRDIFFSOUR CETIMEOUT 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	option	AUTO	1	Use AUTO or SET to set the time	Enum	4	H
		SET	2				
3	timeout	0 to 3600 sec		Specify the time (default=0)	Ulong	4	H+4

2.4.104 PSRDIFFTIMEOUT
Sets maximum age of pseudorange differential data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the maximum age of pseudorange differential correction data to use when operating as a rover station. Received pseudorange differential correction data, older than the specified time, is ignored. This time out period also applies to differential corrections generated from RTK corrections.

> ⓘ The RTCA Standard for SCAT-I stipulates that the maximum age of differential correction messages cannot be greater than 22 seconds. Therefore, for RTCA rover users, the recommended PSRDIFF delay setting is 22.

Abbreviated ASCII Syntax: **Message ID: 1450**

PSRDIFFTIMEOUT delay

Factory Default:

       psrdifftimeout 300

ASCII Example:

       psrdifftimeout 60

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	PRSDIFFTIMEOUT 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	delay	2 to 1000 s		Maximum pseudorange differential age	Ulong	4	H

2.4.105 QZSSECUTOFF
Sets QZSS satellite elevation cutoff

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the elevation cut-off angle for QZSS satellites. The receiver does not start automatically searching for a QZSS satellite until it rises above the cut-off angle (when satellite position is known). Tracked QZSS satellites that fall below the QZSSECUTOFF angle are no longer tracked unless they are manually assigned (see the ASSIGN command on *page 72*).

> ⓘ 1. Use the `ELEVATIONCUTOFF` command (*page 144*) to set the cut-off angle for all other systems.
>
> 2. For the OEM617D and FlexPak6D receivers, this command applies to both the primary and secondary antennas.

Abbreviated ASCII Syntax: **Message ID: 1350**

QZSSECUTOFF angle

Factory Default:

        qzssecutoff 5.0

ASCII Example

        qzssecutoff 10.0

This command permits a negative cut-off angle and can be used in the following situations:

- The antenna is at a high altitude and can look below the local horizon
- Satellites are visible below the horizon due to atmospheric refraction

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	QZSSECUTOFF 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	angle	±90 degrees		Elevation cutoff angle relative to the horizon	Float	4	H

2.4.106 RAIMMODE
Configures RAIM mode

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to configure Receiver Autonomous Integrity Monitoring (RAIM) operation. This command uses RTCA MOPS characteristics which defines the positioning accuracy requirements for airborne lateral navigation (LNAV) and vertical navigation (VNAV) at 3 stages of flight:

1. En route travel
2. Terminal (within range of air terminal)
3. Non-precision approach

In order to ensure that the required level of accuracy is available in these phases of flight, MOPS requires the computation of protection levels (HPL and VPL). MOPS has the following definitions that apply to NovAtel's RAIM feature:

Horizontal Protection Level (HPL) is a radius of the circle in the horizontal plane. Its center is at the true position, that describes the region, assured to contain the indicated horizontal position. It is the horizontal region where the missed alert and false alert requirements are met using autonomous fault detection.

Vertical Protection Level (VPL) is half the length of the segment on the vertical axis. Its center is at the true position, that describes the region, assured to contain the indicated vertical position when autonomous fault detection is used.

Horizontal Alert Limit (HAL) is a radius of the circle in the horizontal plane. Its center is at the true position, that describes the region, required to contain the indicated horizontal position with the required probability.

Vertical Alert Limit (VAL) is half the length of the segment on the vertical axis. Its center is at the true position, that describes the region, required to contain the indicated vertical position with certain probability.

Probability of False Alert (P_{fa}) is a false alert defined as the indication of a positioning failure, when a positioning failure has not occurred (as a result of false detection). A false alert would cause a navigation alert.

Detection strategy

NovAtel's RAIM detection strategy uses the weighted Least-Squares Detection (LSA) method. This method computes a solution using a LSA and is based on the sum of squares of weighted residuals. It is a comparison between a root sum of squares of residuals and a decision threshold to determine a pass/fail decision.

Isolation strategy

NovAtel RAIM uses the maximum residual method. Logically it is implemented as a second part of Fault Detection and Exclusion (FDE) algorithm for LSA detection method. Weighted LSA residuals are standardized individually and the largest residual is compared to a decision threshold. If it is more than the threshold, the observation corresponding to this residual is declared faulty.

Abbreviated ASCII Syntax: **Message ID: 1285**

RAIMMODE mode [hal [val [pfa]]]

Factory Default:

        raimmode default

Input Example:

        raimmode user 100 100 0.01
        raimmode terminal

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	RAIMMODE 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	MODE	See *Table 51, Page 253*			Enum	4	H
3	HAL	5 ≤ HAL ≤ 9999.99		Horizontal alert limit (m) (Default = 0.0)	Double	8	H+4
4	VAL	5 ≤ VAL ≤ 9999.99		Vertical alert limit (m) (Default = 0.0)	Double	8	H+12
5	PFA	$(P_{fa})= 1e^{-7} \leq P_{fa} \leq 0.25$		Probability of false alert (Default = 0.0)	Double	8	H+20

Table 51: RAIM Mode Types

Binary	ASCII	Description
0	DISABLE	Do not do integrity monitoring of least squares solution
1	USER	User will specify alert limits and probability of false alert
2	DEFAULT	Use NovAtel RAIM (default)
3	APPROACH	Default numbers for non-precision approach navigation modes are used - HAL = 556 m (0.3 nm), VAL = 50 m for LNAV/VNAV

Binary	ASCII	Description
4	TERMINAL	Default numbers for terminal navigation mode are used - HAL = 1855 m (1 nm), no VAL requirement
5	ENROUTE	Default numbers for enroute navigation mode are used - HAL = 3710m (2 nm), no VAL requirement

2.4.107 RESET
Performs a hardware reset

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command performs a hardware reset. Following a RESET command, the receiver initiates a cold start boot up. The receiver configuration reverts either to the factory default, if no user configuration was saved or the last SAVECONFIG settings. Refer to the FRESET command on *page 164* and SAVECONFIG command on *page 271*.

The optional delay field is used to set the number of seconds the receiver is to wait before resetting.

Abbreviated ASCII Syntax: **Message ID: 18**

RESET [delay]

Input Example

reset 30

> 1. The RESET command can be used to erase any unsaved changes to the receiver configuration.
>
> 2. Unlike the FRESET command, the RESET command does not erase data stored in the NVM, such as Almanac and Ephemeris data.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RESET 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	delay (0-60)			Seconds to wait before resetting (default = 0)	Ulong	4	H

2.4.108 RTKANTENNA

Specifies L1 phase center (PC) or ARP and enables/disables PC modeling

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use this command to specify whether to use L1 phase center or Antenna Reference Point (ARP) positioning.

You can also decide whether or not to apply phase center variation modeling. If there are any conditions that make a selected mode impossible, the solution status in the position log will indicate an error or warning.

L1 ARP offsets and L2 ARP offsets can be entered using the BASEANTENNAPCO command on *page 86* and THISANTENNAPCO commands on *page 324*. Phase center variation parameters can be entered using the BASEANTENNAPCV command on *page 88* and THISANTENNAPCV command on *page 325*.

Error states occur if either the rover does not have the necessary antenna information entered or the base is not sending sufficient information to work in the requested mode. Some examples of these error conditions are:

- RTCM Types 23 and 24 messages are received from the base and no model is available for the specified base antenna
- Phase center modeling is requested but the base is only sending RTCM Types 3 and 22
- Position reference to the ARP is requested but no rover antenna model is available

Abbreviated ASCII Syntax: **Message ID: 858**

RTKANTENNA posref pcv

Factory Default:

rtkantenna unknown disable

ASCII Example:

rtkantenna arp enable

ⓘ   This command is used for high precision RTK positioning allowing application of antenna offset and phase center variation parameters.

.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKANTENNA 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	posref	L1PC	0	L1 phase center position reference	Enum	4	H
		ARP	1	ARP position reference			
		UNKNOWN	2	Unknown position reference			
3	pcv	DISABLE	0	Disable PCV modelling	Enum	4	H+4
		ENABLE	1	Enable PCV modelling			
4	Reserved				Bool	4	H+8
5	Reserved				Bool	4	H+12

2.4.109 RTKCOMMAND
Resets or sets the RTK filter to default

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to reset the RTK filter or clear any set RTK parameters. The RESET parameter causes the AdVance RTK algorithm to undergo a complete reset, forcing the system to restart the ambiguity resolution calculations. The USE_DEFAULTS action executes the following command: RTKDYNAMICS AUTO.

Abbreviated ASCII Syntax: **Message ID: 97**

RTKCOMMAND action

Factory Default:

rtkcommand use_defaults

ASCII Example:

rtkcommand reset

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKCOMMAND 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	action	USE_DEFAULTS	0	Reset to defaults	Enum	4	H
		RESET	1	Reset RTK filter			

2.4.110 RTKDYNAMICS
Sets the RTK dynamics mode

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to specify how the receiver looks at the data. There are three modes: STATIC, DYNAMIC and AUTO. The STATIC mode forces the RTK software to treat the rover station as though it were stationary.

DYNAMIC mode forces the software to treat the rover as though it were in motion. If the receiver is undergoing very slow, steady motion (<2.5 cm/s for more than 5 seconds), use DYNAMIC mode (as opposed to AUTO) to prevent inaccurate results and possible resets.

> (i) For reliable performance, the antenna should not move more than 1-2 cm when in STATIC mode.

Abbreviated ASCII Syntax: **Message ID: 183**

RTKDYNAMICS mode

Factory Default:

rtkdynamics dynamic

ASCII Example:

rtkdynamics static

Table 52: Dynamics Mode

ASCII	Binary	Description
AUTO	0	Automatically determines dynamics mode
STATIC	1	Static mode
DYNAMIC	2	Dynamic mode

> (i) Use the STATIC option to decrease the time required to fix ambiguities and reduce the amount of noise in the position solution. If STATIC mode is used when the antenna is not static, the receiver will have erroneous solutions and unnecessary RTK resets.

.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKDYNAMICS 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	mode	*Table 52, Page 259*		Set the dynamics mode	Enum	4	H

2.4.111 *RTKELEVMASK*
Sets the RTK elevation mask

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets the elevation mask for generating RTK corrections at a base station. Any satellites below the elevation mask will not be included in the correction messages. Intended for RTCA, it works only with RTCAOBS, RTCAOBS2 or RTCAOBS3 *page 634*. This command is useful when the amount of bandwidth available for transmitting corrections is limited.

Abbreviated ASCII Syntax: **Message ID: 91**

RTKELEVMASK masktype [angle]

Factory Default:

       rtkelevmask auto 0.0

ASCII Example:

       rtkelevmask auto 2.0

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKELEVMASK 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	MaskType	AUTO	0	Type of elevation mask for RTK	Enum	4	H
		USER	1				
3	Angle	0 to 90		Elevation mask angle (default=-1.0)	Float	4	H+4

2.4.112 RTKMATCHEDTIMEOUT
Sets RTK filter reset time after corrections are lost

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets the length of time the receiver continues to use the last RTK correction data once the corrections stop. Once this time is reached, the RTK filter is reset.

Abbreviated ASCII Syntax: **Message ID: 1447**

RTKMATCHEDTIMEOUT timeout

ASCII Example:

rtkmatchedtimeout 180

Factory Default

rtkmatchedtimeout 300

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKMATCHED TIMEOUT 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	timeout	1 to 3600 s		Time out period	Ulong	4	H

2.4.113 RTKNETWORK
Specifies the RTK network mode

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Network RTK uses permanent base station installations, allowing kinematic GNSS users to achieve centimetre accuracies, without the need of setting up a GNSS base station, at a known site. This command sets the RTK network mode for a specific network. For more details on Network RTK, refer to the application note APN-041 Network RTK, available on our website a www.novatel.com/support/search.

Abbreviated ASCII Syntax: **Message ID: 951**

RTKNETWORK mode [network#]

Factory Default:

rtknetwork auto

Input Example:

rtknetwork imax

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTK-NETWORK 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	mode	*Table 53, Page 264*		RTK network mode. The factory default is auto where the receiver switches to the first available network RTK source	Enum	4	H
3	network#	4-24		Specify a number for the network (default = 0)	Ulong	4	H+4

Table 53: Network RTK Mode

Binary	ASCII	Description
0	DISABLE	Single reference station RTK mode. All received network RTK corrections are ignored.
1-4	Reserved	
5	VRS	The Virtual Reference Station (VRS) or Virtual Base Station (VBS) idea introduced by Trimble, is that a base station is artificially created in the vicinity of a rover receiver. All baseline length dependent errors, such as abnormal troposphere variation, ionospheric disturbances and orbital errors, are reduced for this VRS. The rover receiving VRS information has a lower level of these errors than a distant base station. The VRS is calculated for a position, supplied by the rover during communication start up, with networking software. The VRS position can change if the rover is far away from the initial point. The format for sending the rover's position is standard NMEA format. Most rovers receive VRS data, for a calculated base station, within a couple of metres away. The VRS approach requires bi-directional communication for supplying the rover's position to the networking software.
6	IMAX	The iMAX idea, introduced by Leica Geosystems, is that networking software corrections, based on the rover's position, are calculated as with VRS. However, instead of calculating the base station observations for the provided position or another position closer to the base station, original observation information is corrected with the calculated corrections and broadcast. VRS works so that although the rover is unaware of the errors the VRS is taking care of, there still might be ionospheric remains in the base station observations. iMAX provides actual base station position information. The rover may assume the base station is at a distance and open its settings for estimation of the remaining ionospheric residuals. The iMAX method may trigger the rover to open its settings further than required, since the networking software removes at least part of the ionospheric disturbances. However, compared to VRS above, this approach is safer since it notifies the rover when there might be baseline length dependent errors in the observation information. iMAX requires bi-directional communication to the networking software for supplying the base station observation information.
7	FKP	The FKP method delivers the information from a base station network to the rover. No precise knowledge of the rover's position is required for providing the correct information. The corrections are deployed as gradients to be used for interpolating to the rover's actual position.
8	MAX	The basic principle of the master-auxiliary concept is to provide, in compact form, as much of the information from the network and the errors it is observing to the rover as possible. With more information about the state and distribution of the dispersive and non-dispersive errors across the network, the rover is able to use more intelligent algorithms in the determination of its position solution. Each supplier of reference station software will have their own proprietary algorithms for modeling or estimating these error sources. The rover system can decide to use or to neglect the network RTK information, depending on its own firmware algorithm performance.
9	Reserved	
10	AUTO	Default value, assume single base. If network RTK corrections are detected then the receiver will switch to the appropriate mode. iMAX and VRS can only be detected using RTCMV3, however, it is not possible to distinguish between iMAX or VRS. If iMAX or VRS is detected, then iMAX will be assumed.

2.4.114 RTKQUALITYLEVEL
Sets an RTK quality mode

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use this command to select an RTK quality mode.

Abbreviated ASCII Syntax: **Message ID: 844**

RTKQUALITYLEVEL mode

Factory Default:

rtkqualitylevel normal

ASCII Example:

rtkqualitylevel extra_safe

Table 54: RTK Quality Mode

ASCII	Binary	Description
NORMAL	1	Normal RTK
EXTRA_SAFE	4	Extra Safe RTK

ⓘ The EXTRA_SAFE command is needed in areas where the signal is partially blocked and the position solution in NORMAL mode shows NARROW_INT even though the real position solution is out by several metres. Using EXTRA_SAFE in these environments means the solution will be slower getting to NARROW_INT but it is less likely to be erroneous.

.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKQUALITY-LEVEL 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	mode	See *Table 54, Page 265*		Set the RTK quality level mode	Enum	4	H

2.4.115 RTKSOURCE
Sets the RTK correction source

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to identify from which base station to accept RTK (RTCM, RTCMV3, RTCA, CMR and OmniSTAR (HP/XP)) differential corrections. This is useful when the receiver is receiving corrections from multiple base stations. See also the PSRDIFFSOURCE command on *page 246*.

> (i) To use OmniSTAR HP/XP differential corrections, a NovAtel receiver with L-Band capability and a subscription to the OmniSTAR service are required. Contact NovAtel for details.

Abbreviated ASCII Syntax: **Message ID: 494**

RTKSOURCE type [id]

Factory Default:

> rtksource auto any

ASCII Examples:

1. Specify the format before specifying the base station IDs:

   > rtksource rtcmv3 5
   > rtksource rtcm 6

2. Select only SBAS:

   > rtksource none
   > psrdiffsource sbas
   > sbascontrol enable auto

3. Enable OmniSTAR HP and VBS:

   > rtksource omnistar
   > psrdiffsource omnistar

4. Enable RTK and PSRDIFF from RTCM, with a fall-back to SBAS:

   > rtksource rtcm any
   > psrdiffsource rtcm any
   > sbascontrol enable auto

> (i) Consider an agricultural example where a farmer has their own RTCM base station set up but due to either obstructions or radio problems, occasionally experiences loss of corrections. By specifying a fall back to SBAS, the farmer could set up their receiver to use transmitted RTCM corrections when available but fall back to SBAS. Also, if they decided to get an OmniSTAR subscription, they could switch to the OmniSTAR corrections.

.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKSOURCE 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	type	See *Table 50, Page 248*	ID Type [a]	Enum	4	H
3	Base station ID	Char [4] or ANY	ID string	Char[5]	8 [b]	H+4

a. If ANY chosen, the receiver ignores the ID string. Specify a type when using base station IDs.
b. In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.

2.4.116 RTKSOURCETIMEOUT
Sets RTK correction source timeout

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

When multiple RTK correction sources are available, this command allows the user to set a time, in seconds, that the receiver will wait before switching to another RTK correction source if corrections from the original source are lost.

Abbreviated ASCII Syntax: **Message ID: 1445**

RTKSOURCETIMEOUT option [timeout]

Factory Default:

 rtksourcetimeout auto

ASCII Example:

 rtksourcetimeout auto
 rtksourcetimeout set 180

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKSOURCE TIMEOUT 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	option	AUTO[a]	1	Timeout field is optional for AUTO and has no effect	Enum	4	H
		SET	2	0 is not accepted by SET command			
3	timeout	1 to 3600 s (maximum)		Specify the time (default=0 for the AUTO option)	Ulong	4	H+4

 a. AUTO option sets timeout according to network type or other self-detected conditions.

2.4.117 RTKSVENTRIES
Sets number of satellites in corrections

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets the number of satellites (at the highest elevation) that are transmitted in the RTK corrections from a base station receiver. Intended for RTCA, it works only with RTCAOBS, RTCAOBS2 or RTCAOBS3 (*page 634*). This is useful when the amount of bandwidth available for transmitting corrections is limited.

Abbreviated ASCII Syntax: **Message ID: 92**

RTKSVENTRIES number

Factory Default:

rtksventries 24

ASCII Example:

rtksventries 7

> (i) GPS devices have enabled many transit and fleet authorities to provide Automatic Vehicle Location (AVL). AVL systems track the position of individual vehicles and relay that data back to a remote dispatch location that can store or better utilize the information. Consider the implementation of an AVL system within a police department, to automatically log and keep track of the location of each cruiser. Typically a fleet uses a 9600 bps connection where AVL data is relayed back to headquarters. The limited bandwidth of the radio must be shared amongst the AVL and other systems in multiple cruisers.
>
> When operating with a low baud rate radio transmitter (9600 or lower), especially over a long distance, the AVL system could limit the number of satellites for which corrections are sent using the RTKSVENTRIES command.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKSVENTRIES 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	number	4-24		The number of SVs to be transmitted in correction messages	Ulong	4	H

2.4.118 RTKTIMEOUT
Sets maximum age of RTK data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

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.

Abbreviated ASCII Syntax: **Message ID: 910**

RTKTIMEOUT delay

Factory Default:

> rtktimeout 60

ASCII Example (rover):

> rtktimeout 20

> (i) See the `DGPSEPHEMDELAY` command on *page 124* to set the ephemeris changeover delay for base stations.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKTIMEOUT 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	delay	5 to 60 s		Maximum RTK data age	Ulong	4	H

2.4.119 *SAVECONFIG*
Save current configuration in NVM

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command saves the present configuration in Non-Volatile Memory (NVM). The configuration includes the current log settings, FIX settings, port configurations and so on. The output is in the RXCONFIG log, see *page 657*. See also the FRESET command, *page 164*.

> If using the SAVECONFIG command in NovAtel Connect, ensure that you have all windows other than the Console window closed. Otherwise, log requests used for the various windows are saved as well. This will result in unnecessary data being logged.

Abbreviated ASCII Syntax: **Message ID: 19**

SAVECONFIG

2.4.120 SAVEETHERNETDATA
Save the configuration data associated with an Ethernet interface

OEM Platform: 628, 638, FlexPak6, ProPak6

Saving the configuration data for an Ethernet interface allows the interface to start automatically at boot time and be configured with either a static IP address or to obtain an address using DHCP. The SAVEETHERNETDATA command (*page 272*) saves the configuration for the interface entered using the ETHCONFIG, IPCONFIG and DNSCONFIG commands (pages *146*, *188* and *132*). The ETHCONFIG and IPCONFIG commands must have been invoked before the SAVEETHERNETDATA command is used. The configuration data that is saved will survive a RESET and FRESET command (*page 164*). To clear the Ethernet interface configuration data, the FRESET ETHERNET command is used. It is not necessary to issue the SAVECONFIG command to save the Ethernet interface configuration data. In fact, if you use SAVECONFIG to save the ETHCONFIG, IPCONFIG and DNSCONFIG commands, the configuration saved by SAVEETHERNETDATA will take precedence over the SAVECONFIG configuration.

Abbreviated ASCII Syntax: **Message ID: 1679**

SAVEETHERNETDATA [interface]

ASCII Example:

      ethconfig etha auto auto auto auto
      ipconfig etha static 192.168.8.11 255.255.255.0 192.168.8.1
      dnsconfig 1 192.168.4.200
      saveethernetdata etha

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	SAVEETHERNET DATA 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	Interface	ETHA	2	The Ethernet interface to save the configuration data for. The default is ETHA.	Enum	4	H

Note that the configurations set using the ICOMCONFIG and NTRIPCONFIG commands are not saved by the SAVEETHERDATA command. The following factory default ICOM configurations can be used if Ethernet access to the receiver is required immediately after the receiver is RESET or FRESET.

      icomconfig icom1 tcp :3001
      icomconfig icom2 tcp :3002
      icomconfig icom3 tcp :3003

See also the following commands: ETHCONFIG (*page 146*), IPCONFIG (*page 191*), DNSCONFIG (page *page 132*) and FRESET (*page 164*).

2.4.121 SBASCONTROL
Sets SBAS test mode and PRN

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to dictate how the receiver tracks and uses correction data from Satellite Based Augmentation Systems (SBAS).

To enable the position solution corrections, issue the SBASCONTROL ENABLE command. The receiver does not, by default, attempt to track or use any SBAS signals satellites unless told to do so by the SBASCONTROL command.When in AUTO mode, if the receiver is outside the defined satellite system's corrections grid, it reverts to ANY mode and chooses a system based on other criteria.

Once tracking satellites from one system in ANY or AUTO mode, it does not track satellites from other systems. This is because systems such as WAAS, EGNOS and MSAS do not share broadcast information and have no way of knowing the others are there.

The "testmode" parameter in the example provides a method to use a particular satellite even if it is currently operating in test mode. The recommended setting for tracking satellites operating in test mode is ZEROTOTWO. On a simulator, you may want to leave this parameter off or specify NONE explicitly.

When using the SBASCONTROL command to direct the receiver to use a specific correction type, the receiver begins to search for and track the relevant GEO PRNs for that correction type only.

The receiver can be forced to track a specific PRN using the ASSIGN command. The receiver can also be forced to use the corrections from a specific SBAS PRN using the SBASCONTROL command.

Disable stops the corrections from being used.

Abbreviated ASCII Syntax: **Message ID: 652**

SBASCONTROL switch [system] [prn] [testmode]

Factory Default:

> sbascontrol disable

ASCII Example:

> sbascontrol enable waas

Table 55: System Types

ASCII	Binary	Description
NONE	0	Does not use any SBAS satellites
AUTO	1	Automatically determines satellite system to use and prevents the receiver from using satellites outside of the service area (default)
ANY	2	Uses any and all SBAS satellites found
WAAS	3	Uses only WAAS satellites
EGNOS	4	Uses only EGNOS satellites
MSAS	5	Uses only MSAS satellites
GAGAN	6	Uses only GAGAN satellites
QZSS	7	Uses only QZSS SAIF signals

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SBASCONTROL 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	switch	DISABLE	0	Receiver does not use the SBAS corrections it receives (default)	Enum	4	H
		ENABLE	1	Receiver uses the SBAS corrections it receives			
3	system	See *Table 55, System Types* on *page 274*		Choose the SBAS the receiver will use (default = AUTO)	Enum	4	H+4
4	prn	0		Receiver uses any PRN (default)	Ulong	4	H+8
		120-138		Receiver uses SBAS corrections only from this PRN			
5	testmode	NONE	0	Receiver interprets Type 0 messages as they are intended (as do not use) (default)	Enum	4	H+12
		ZEROTO TWO	1	Receiver interprets Type 0 messages as Type 2 messages			
		IGNORE ZERO	2	Receiver ignores the usual interpretation of Type 0 messages (as do not use) and continues			

2.4.122 SBASECUTOFF
Sets SBAS satellite elevation cut-off

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets the elevation cut-off angle for SBAS satellites. The receiver does not start automatically searching for an SBAS satellite until it rises above the cut-off angle (when satellite position is known). Tracked SBAS satellites that fall below the SBASECUTOFF angle are no longer tracked unless they are manually assigned (see the `ASSIGN` command on *page 72*).

> ⓘ Use the `ELEVATIONCUTOFF` command (*page 144*) to set the cut-off angle for all other systems.

Abbreviated ASCII Syntax: **Message ID: 1000**

SBASECUTOFF angle

Factory Default:

        sbasecutoff -5.0

ASCII Example:

        sbasecutoff 10.0

> ⓘ This command permits a negative cut-off angle and can be used in the following situations:
> - The antenna is at a high altitude and can look below the local horizon
> - Satellites are visible below the horizon due to atmospheric refraction

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SBASECUTOFF 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	angle	±90.0 degrees		Elevation cut-off angle relative to horizon	Float	4	H

2.4.123 SBASTIMEOUT
Sets the SBAS position time out

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to set the amount of time the receiver remains in an SBAS position if it stops receiving SBAS corrections.

Abbreviated ASCII Syntax: **Message ID: 1001**

SBASTIMEOUT mode [delay]

Factory Default:

       sbastimeout auto

ASCII Example:

       sbastimeout set 100

> (i) When the time out mode is AUTO, the time out delay is 180 s.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SBASTIMEOUT 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	mode	See *Table 56, Page 276* below		Time out mode	Enum	4	H
3	delay	2 to 1000 s		Maximum SBAS position age (default=180)	Double	8	H+4
4	Reserved				Double	8	H+12

Table 56: SBAS Time Out Mode

Binary	ASCII	Description
0		Reserved
1	AUTO	Set the default value (180 s)
2	SET	Set the delay in seconds

2.4.124 SELECTCHANCONFIG
Sets the channel configuration

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Some software models come with support for more than one channel configuration, which can be verified by logging CHANCONFIGLIST (see *page 430*). The SELECTCHANCONFIG command is used to pick a different channel configuration. If a different channel configuration is selected via the SELECTCHANCONFIG command, the receiver resets and starts up with the new configuration. The Set in Use number in the CHANCONFIGLIST log changes as a result.

> (i) After a FRESET, the channel configuration is reset to 1.

Abbreviated ASCII Syntax: **Message ID: 1149**

SELECTCHANCONFIG chanconfigsetting

Factory Default:

selectchanconfig 1

ASCII Example:

selectchanconfig 2

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	SELECTCHANCONFIG 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	chanconfigsetting	Channel configuration to use	Ulong	4	H

(i) Below is a use case example of the SELECTCHANCONFIG command. Abbreviated ASCII commands and logs are used to better illustrate the example.

1. LOG VERSION to show the current model. For example:

   GPSCARD "D2LR0RCCR" "BFN11230026" "OEM628-1.00" "OEM060200RN0000" "OEM060200RB0000" "2012/MAR/22" "10:51:30"

2. "D2L" prefix in the model indicates it is a GPS+GLONASS L1/L2 with SBAS and LBAND functionality.

3. LOG CHANCONFIGLIST to show what the channel configuration options are and which channel configuration set is being used.

   CHANCONFIGLIST COM1 0 84.5 FINESTEERING 1684 158935.214 00000020 D1C0 10526

     1 2
     4
     16 GPSL1L2
     4 SBASL1
     14 GLOL1L2
     1 LBAND
     4
     16 GPSL1L2PL2C
     4 SBASL1
     14 GLOL1L2PL2C
     1 LBAND

4. There are two options given for the model and the first channel configuration set is currently being used.

5. If the user would like to use the second channel configuration set where the receiver would explicitly track both L2P and L2C signals, SELECTCHANCONFIG 2 would be entered.

6. The receiver receives the command and resets. At startup, the second channel configuration set is configured.

7. To verify that setting has changed, enter LOG CHANCONFIGLIST

   CHANCONFIGLIST COM1 0 84.5 FINESTEERING 1684 158935.214 00000020 D1C0 10526

     4
     16 GPSL1L2PL2C
     4 SBASL1
     14 GLOL1L2PL2C
     1 LBAND

8. To further verify, enter LOG TRACKSTAT to show all the configured channels.

2.4.125 SEND
Sends an ASCII message to a COM port

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to send ASCII printable data from any of the COM or USB ports to a specified communications port. This is a one time command, therefore the data message must be preceded by the SEND command and followed by <CR> each time data is sent. If the data string contains delimiters (that is, spaces, commas, tabs and so on), the entire string must be contained within double quotation marks. Carriage return and line feed characters (for example, 0x0D, 0x0A) are appended to the sent ASCII data.

Abbreviated ASCII Syntax: **Message ID: 177**

SEND [port] data

ASCII Example
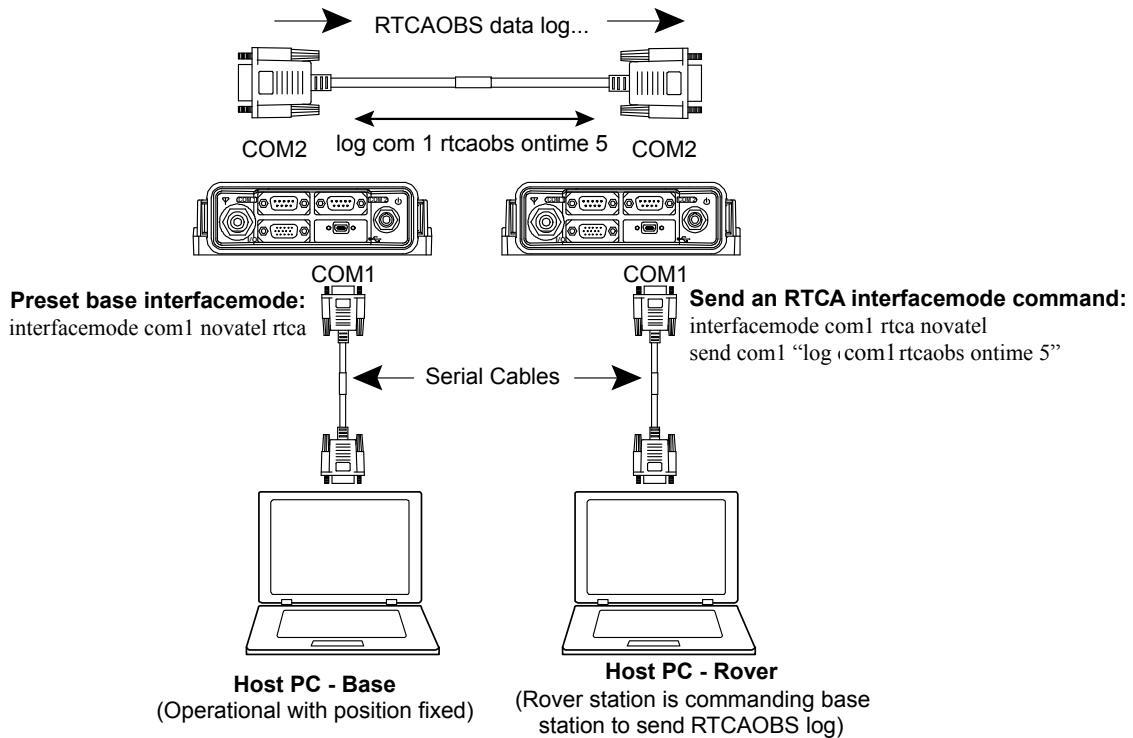
send com1 "log com1 rtcaobs ontime 5"

> ⓘ **Scenario:** Assume you are operating receivers as base and rover stations. It could also be assumed that the base station is unattended but operational and you wish to control it from the rover station. From the rover station, you could establish the data link and command the base station receiver to send differential corrections.

> **PPk6** The SEND command can be used to send commands and configure the OEM615 within the ProPak6 via COM5.

Figure 8: Using the SEND Command

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SEND 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	See *Table 4, Page 27*		Output port (default=THISPORT)	Enum	4	H
3	message	Max 100 character string (99 typed visible chars and a null char added by the firmware automatically)		ASCII data to send	String [max 100]	Variable[a]	H+4

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.126 SENDHEX
Send non-printable characters in hex pairs

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is like the `SEND` (*page 279*) command except it is used to send non-printable characters expressed as hexadecimal pairs. Carriage return and line feed characters (for example, 0x0D, 0x0A) will **not** be appended to the sent data and so must be explicitly added to the data if needed.

Abbreviated ASCII Syntax:	**Message ID: 178**

SENDHEX port length data

Input Example:

sendhex com1 6 143ab5910d0a

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SENDHEX 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	See *Table 4, Page 27*		Output port (default=THISPORT)	Enum	4	H
3	length	0 - 700		Number of hex pairs	Ulong	4	H+4
4	message	limited to a 700 maximum string (1400 pair hex). Even number of ASCII characters from set of 0-9, A-F. No spaces are allowed between pairs of characters		Data	String [max 700]	Variable[a]	H+8

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.127 SERIALCONFIG
Configures serial port settings

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

ⓘ
1. This replaces the `COM` command. Also refer to the `ECHO` command on *page 140*.

2. The `SERIALCONFIG` command can be used as a log to confirm settings.

3. The entire content of the current log is sent before pausing due to the receipt of the XOFF character.

This command is used to configure the receiver's asynchronous serial port communications drivers. The current SERIALCONFIG port configuration can be reset to its default state at any time by sending it two hardware break signals of 250 milliseconds each, spaced by fifteen hundred milliseconds (1.5 seconds) with a pause of at least 250 milliseconds following the second break. This will:

- Stop the logging of data on the current port (see UNLOGALL on *page 342)*
- Clear the transmit and receive buffers on the current port
- Return the current port to its default settings (see *Factory Defaults* on *page 41* for details)
- Set the interface mode to NovAtel for both input and output (see the `INTERFACEMODE` command on *page 184*)

See also *Factory Defaults* on *page 41* for a description of the factory defaults.

ⓘ
1. The `COMCONTROL` command, see *page 113*, may conflict with handshaking of the selected COM port. If handshaking is enabled, then unexpected results may occur.

2. Baud rates higher than 115,200 bps are not supported by standard PC hardware. Special PC hardware may be required for higher rates, including 230400 bps, 460800 bps and 921600 bps. Also, some PC's have trouble with baud rates beyond 57600 bps. Avoid having COM ports of two receivers connected together using baud rates that do not match. Data transmitted through a port operating at a slower baud rate may be misinterpreted as break signals by the receiving port if it is operating at a higher baud rate because data transmitted at the lower baud rate is stretched relative to the higher baud rate. In this case, configure the receiving port to break detection disabled using the `SERIALCONFIG` command.

🛑
Use the `SERIALCONFIG` command before using the `INTERFACEMODE` command on each port. Turn break detection off using the `SERIALCONFIG` command to stop the port from resetting because it is interpreting incoming bits as a break command.

Abbreviated ASCII Syntax: **Message ID: 1246**

SERIALCONFIG [port] baud [parity[databits[stopbits[handshaking[break]]]]]

Factory Defaults:

serialconfig com1 9600 n 8 1 n on   serialconfig com2 9600 n 8 1 n on
serialconfig com3 9600 n 8 1 n on   serialconfig com4 9600 n 8 1 n on
serialconfig com5 9600 n 8 1 n on   serialconfig com6 9600 n 8 1 n on

ASCII Example:

serialconfig com1 9600 n 8 1 n off

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	SERIALCONFIG 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	See *Table 57, COM Port Identifiers* on *page 284*		Port to configure (default = THISPORT)	Enum	4	H
3	bps/baud	300, 600, 900, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800 and 921600		Communication baud rate (bps).	Ulong	4	H+4
4	parity	See *Table 58, Parity* on *page 284*		Parity	Enum	4	H+8
5	databits	7 or 8		Number of data bits (default = 8)	Ulong	4	H+12
6	stopbits	1 or 2		Number of stop bits (default = 1)	Ulong	4	H+16
7	handshake[a]	See *Table 59, Handshaking* on *page 284*		Handshaking	Enum	4	H+20
8	break	OFF	0	Disable break detection	Enum	4	H+24
		ON	1	Enable break detection (default)			

a. On the OEM615 and OEM628, The AUX and COM3 ports do not support hardware handshaking. Only transmit and receive lines exist for these ports.

Table 57: COM Port Identifiers

Binary	ASCII	Description	Applicable Card
1	COM1	COM port 1	OM615, OEM628, OM638
2	COM2	COM port 2	OM615, OEM628, OM638
3	COM3	COM port 3	OM615, OEM628, OM638
6	THISPORT	The current COM port	OM615, OEM628, OM638
7	FILE	For logging to a file	OEM638
8	ALL	All COM ports	OM615, OEM628, OM638
19	COM4	COM port 4	OEM638
21	IMU	IMU COM port	dependent on hardware configuration
31	COM5	COM port 5	OEM638
32	COM6	COM port 6	OEM638
33	BT1	Bluetooth COM port	dependent on hardware configuration
34	COM7	COM port 7	ProPak6 only via expansion cable
35	COM8	COM port 8	ProPak6 only via expansion cable
36	COM9	COM port 9	ProPak6 only via expansion cable
37	COM10	COM port 10	ProPak6 only via expansion cable

Table 58: Parity

Binary	ASCII	Description
0	N	No parity (default)
1	E	Even parity
2	O	Odd parity

Table 59: Handshaking

Binary	ASCII	Description
0	N	No handshaking (default)
1	XON	XON/XOFF software handshaking
2	CTS	CTS/RTS hardware handshaking

2.4.128 SERIALPROTOCOL
Sets the protocol to be used by a serial port

OEM Platform: 628, 638, FlexPak6, ProPak6

On some OEM6 receiver cards, selected ports can support either RS-232 or RS-422 signaling protocol. The default protocol is RS-232. The SERIALPROTOCOL command is used to select the protocol (RS-232 or RS-422) supported on the port.

ⓘ In some cases, the protocol used on a serial port can also be selected at boot time by pulling a USER_IO line high or low, as appropriate. See the OEM6 Family Installation and Operation User Manual (OM-20000128) for more information

The SERIALPROTOCOL command can be used to override the protocol selected at boot time.

ⓘ On the OEM615, there are no COM ports that support the RS-422 protocol.

Abbreviated ASCII Syntax: **Message ID: 1444**

SERIALPROTOCOL port protocol

ASCII Example:

serialprotocol com2 rs422

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	SERIAL PROTOCOL 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	See *Table 60, Page 285*		Select the COM port on which the protocol is being set. The port that can be entered depends on the hardware platform being used.	Enum	4	H
3	protocol	RS232	0	Set the port to use RS-232 protocol	Enum	4	H+4
		RS422	1	Set the port to use RS-422 protocol			

Table 60: Ports Supporting RS-422

OEM6 Receiver Type	Allowable Ports	Binary Value
OEM615, 617, 617D, FlexPak6D	None	
OEM628, FlexPak6	COM1	1
OEM638, ProPak6	COM1	1
	COM2	2
	COM6	6

2.4.129 SETADMINPASSWORD
Sets the administration password

OEM Platform: 628, 638, FlexPak6, ProPak6

This command sets the administration password used to log into various web services.

- The administration password is required for FTP access (no guest access).

The default password is the receiver's PSN. This password should be changed before connecting the receiver to a network.

Abbreviated ASCII Syntax: **Message ID: 1579**

SETADMINPASSWORD oldpassword newpassword

Input example

setadminpassword abc123 xyz789

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	SETADMINPASSWORD 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	OldPassword	Maximum 28 character string		Previous password.	String [28]	variable[a]	H
3	NewPassword	Maximum 28 character string		New password.	String [28]	variable[a]	variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

> (i) This password can be restored to default (the receiver's PSN) by issuing the
> FRESET USER_ACCOUNTS command (see *page 164*).

2.4.130 SETAPPROXPOS
Sets an approximate position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets an approximate latitude, longitude and height in the receiver. Estimating these parameters, when used in conjunction with an approximate time (see the SETAPPROXTIME command on *page 288*), can improve satellite acquisition times and Time To First Fix (TTFF). For more information about TTFF and Satellite Acquisition, refer to our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

The horizontal position entered should be within 200 km of the actual receiver position. The approximate height is not critical and can normally be entered as zero. If the receiver cannot calculate a valid position within 2.5 minutes of entering an approximate position, the approximate position is ignored.

The approximate position is not visible in any position logs. It can be seen by issuing a SETAPPROXPOS log. See also the SATVIS log on *page 670*.

Abbreviated ASCII Syntax: **Message ID: 377**

SETAPPROXPOS lat lon height

Input Example:

setapproxpos 51.116 -114.038 0

> (i) For an example on the use of this command, refer to the SETAPPROXTIME command on *page 288*.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETAPPROXPOS 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	Lat	± 90 degrees		Approximate latitude	Double	8	H
3	Lon	± 180 degrees		Approximate longitude	Double	8	H+8
4	Height	-1000 to +20000000 m		Approximate height	Double	8	H+16

2.4.131 SETAPPROXTIME
Sets an approximate GPS reference time

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets an approximate time in the receiver. The receiver uses this time as system time until a coarse time can be acquired. This can be used in conjunction with an approximate position (see the SETAPPROXPOS command on *page 287*) to improve Time To First Fix (TTFF). For more information TTFF and Satellite Acquisition, refer to our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

The time entered should be within 10 minutes of the actual GPS reference time. If the week number entered does not match the broadcast week number, the receiver resets once it is tracking.

Abbreviated ASCII Syntax: **Message ID: 102**

SETAPPROXTIME week sec

Input Example:

setapproxtime 1605 425384

> Upon power up, the receiver does not know its position or time and therefore cannot use almanac information to aid satellite acquisition. You can set an approximate GPS reference time using the SETAPPROXTIME command or RTCAEPHEM message. The RTCAEPHEM message contains GPS reference week and seconds and the receiver uses that GPS reference time if the time is not yet known. Several logs provide base station coordinates and the receiver uses them as an approximate position allowing it to compute satellite visibility. Alternately, you can set an approximate position by using the SETAPPROXPOS command (*page 287*).
>
> Approximate time and position may be used in conjunction with a current almanac to aid satellite acquisition. See the table below for a summary of the OEM6 family commands and logs used to inject an approximated time or position into the receiver:
>
> | Approximate | Command | Log |
> |---|---|---|
> | Time | SETAPPROXTIME | RTCAEPHEM |
> | Position | SETAPPROXPOS | RTCAREF *or* CMRREF *or* RTCM3 |
>
> Base station aiding can help in these environments. A set of ephemerides can be injected into a rover station by broadcasting the RTCAEPHEM message from a base station. This is also useful in environments where there is frequent loss of lock. GPS ephemeris is three frames long within a sequence of five frames. Each frame requires 6 s of continuous lock to collect the ephemeris data. This gives a minimum of 18 s and a maximum of 36 s continuous lock time or when no recent ephemerides (new or stored) are available. See also the SATVIS log on *page 670*.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETAPPROXTIME 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	week	0-9999		GPS reference week number	Ulong	4	H
3	sec	0-604800		Number of seconds into GPS reference week	Double	8	H+4

2.4.132 SETBASERECEIVERTYPE
Sets base receiver type

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command allows the user to specify the base receiver type to aid GLONASS ambiguity fixing in RTK. It can be used as a substitute for RTCM1033 messages that contains the information on the base receiver type. This command should be issued to the Rover. Using this allows the Rover receiver to fix the GLONASS ambiguities when using RTCM2.X and CMR corrections as well.

> An incorrect base type setting can significantly impair ambiguity resolution.

Abbreviated ASCII Syntax: **Message ID: 1374**

SETBASERECEIVERTYPE base_type

Factory Default:

setbasereceivertype unknown

ASCII Example:

setbasereceivertype novatel

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SBASERECEIVER TYPE 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	base_type	unknown	0	Unknown Base	-	4	H
		novatel	1	NovAtel Base			
		trimble	2	Trimble Base			
		topcon	3	Topcon Base			
		magellan	4	Magellan Base			
		leica	5	Leica Base			

2.4.133 SETBESTPOSCRITERIA
Sets selection criteria for BESTPOS

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use this command to set the criteria for the BESTPOS log and choose between 2D and 3D standard deviation to obtain the best position from the BESTPOS log. It also allows you to specify the number of seconds to wait before changing the position type. This delay provides a single transition that ensures position types do not skip back and forth. See also BESTPOS log on *page 406*.

> (i) The SETBESTPOSCRITERIA command is also used as the basis for the UALCONTROL (*page 331*) standard deviations.

Abbreviated ASCII Syntax: **Message ID: 839**

SETBESTPOSCRITERIA type [delay]

Factory Default:

       setbestposcriteria pos3d 0

Input Example:

       setbestposcriteria pos2d 5

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	SETBESTPOS CRITERIA 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	type	See *Table 61, Page 291*		Select a 2D or 3D standard deviation type to obtain the best position from the BESTPOS log	Enum	4	H
3	delay	0 to 100 s		Set the number of seconds to wait before changing the position type. Default=0	Ulong	4	H+4

Table 61: Selection Type

ASCII	Binary	Description
POS3D	0	3D standard deviation
POS2D	1	2D standard deviation

2.4.134 SETCANNAME
Sets the CAN name fields

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, ProPak6

This commands sets the CAN device name fields.

Abbreviated ASCII Syntax: **Message ID: 1091**

SETCANNAME manufacturercode [industrygroup] [deviceclass] [deviceclassinstance] [function] [functioninstance] [ecuinstance] [preferredaddress]

Input Example:

setcanname 305

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	SETCANNAME 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	ManufacturerCode			CAN module's Manufacturer Code	Ulong	4	H
3	IndustryGroup			Industry group number (default = 2)	Ulong	4	H+4
4	DeviceClass			11783-5 Device class (default = 0)	Ulong	4	H+8
5	DeviceClassInstance			11783-5 Device class instance (default = 0)	Ulong	4	H+12
6	Function			11783-5 Function (default = 23)	Ulong	4	H+16
7	FunctionInstance			11783-5 Function instance (default = 0)	Ulong	4	H+20
8	ECUInstance			11783-5 ECU Instance (default = 0)	Ulong	4	H+24
9	PreferredAddress			Device default address on start up (default=28)	Ulong	4	H+28
10	Reserved				Ulong	4	H+32

2.4.135 SETDIFFCODEBIASES
Sets satellite differential code biases

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

> Changing the biases may negatively affect positioning accuracy. NovAtel recommends that only advanced users modify the biases.

Use this command to set the differential code biases that correct pseudorange errors affecting the L1/L2 ionospheric corrections. Bias values are restricted to between -10 ns and +10 ns. A set of biases is included in the firmware and use of the biases is enabled by default. See also the DIFFCODEBIASCONTROL command on *page 128.*

The receiver uses the C/A code on L1 and the P code on L2 to calculate a dual-frequency ionospheric correction. However, the GNSS clock corrections are broadcast as if the P codes on both L1 and L2 are used to calculate this correction. The biases account for the differences between the P and C/A codes on L1 and improve the estimate of the ionospheric correction.

The biases are calculated by the International GNSS Service (IGS). Calculation details, analysis and results are available at http://aiuws.unibe.ch/spec/dcb.php. The most recent 30 day average bias values can be downloaded from http://aiuws.unibe.ch/ionosphere/p1c1.dcb.

Abbreviated ASCII Syntax: **Message ID: 687**

SETDIFFCODEBIASES bias_type biases

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETDIFFCODE BIASES 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	bias_type	GPS_C1P1 (default)	0	Code pair to which biases refer	Enum	4	H
		GPS_C2P2	1				
3	biases	-10 to +10 ns		Array of 40 biases (ns)	Float[40]	160	H+4

2.4.136 *SETFILECOPYMODE*
Configures the internal memory copy function

OEM Platform: 638, ProPak6

The SETFILECOPYMODE command is used to configure the copy function from internal memory to USB stick. The copy function is only performed if there is no log file open for writing at the time the USB stick is inserted. Subsequently, closing the log file or issuing this command after inserting the USB stick does not initiate the auto-copy procedure. This command has nothing to do with the manual copy procedure (DOSCMD COPY).

> ⓘ To use the auto copy feature, the USB memory stick must be empty to avoid any naming collisions.

Abbreviated ASCII Syntax: **Message ID: 1581**

SETFILECOPYMODE filecopymodeenum [reserved]

Factory Default:

setfilecopymode manual

ASCII Example:

setfilecopymode auto_all

> ⓘ The SETFILECOPYMODE command copies all recorded log files from internal memory to a USB stick. The USB stick must be empty and have the capacity to hold the data. The SETFILECOPYMODE command must be issued prior to inserting the USB stick to trigger auto-copy.

.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETFILECOPYMODE 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	FileCopyModeEnum	MANUAL	0	Manual copy using the DOSCMD COPY function	Enum	4	H
		AUTO_ALL	1	Automatically copy all files			
3	Reserved						

2.4.137 SETIONOTYPE
Enables ionospheric models

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use this command to set which ionospheric corrections model the receiver should use. If the selected model is not available, the receiver reverts to AUTO.

> ⓘ L1 only models automatically use SBAS ionospheric grid corrections, if available.

Abbreviated ASCII Syntax: **Message ID: 711**

SETIONOTYPE model

Factory Default:

setionotype auto

ASCII Example:

setionotype klobuchar

> ⓘ An ionotype of AUTO is recommended for PDP and GLIDE.

.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETIONO-TYPE 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	model	See *Table 62, Page 295*		Choose an ionospheric corrections model	Enum	4	H

Table 62: Ionospheric Correction Models

ASCII	Binary	Description
NONE	0	Don't use ionosphere modelling
KLOBUCHAR	1	Use the Klobuchar model broadcast by GPS
GRID	2	Use the SBAS grid model
L1L2	3	Use the L1/L2 model
AUTO	4	Automatically determine the ionospheric model to use

2.4.138 SETNAV
Sets start and destination waypoints

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command permits entry of one set of navigation waypoints (see *Figure 9*). The origin (FROM) and destination (TO) waypoint coordinates entered are considered on the ellipsoidal surface of the current datum (default WGS84). Once SETNAV has been set, monitor the navigation calculations and progress by observing the NAVIGATE log messages (*page 545*).

Track offset is the perpendicular distance from the great circle line drawn between the FROM lat-lon and TO lat-lon waypoints. It establishes the desired navigation path or track, that runs parallel to the great circle line, which now becomes the offset track, and is set by entering the track offset value in metres. A negative track offset value indicates that the offset track is to the left of the great circle line track. A positive track offset value (no sign required) indicates the offset track is to the right of the great circle line track (looking from origin to destination). See *Figure 9* on *Page 297* for clarification.

Abbreviated ASCII Syntax: **Message ID: 162**

SETNAV fromlat fromlon tolat tolon trackoffset from-point to-point
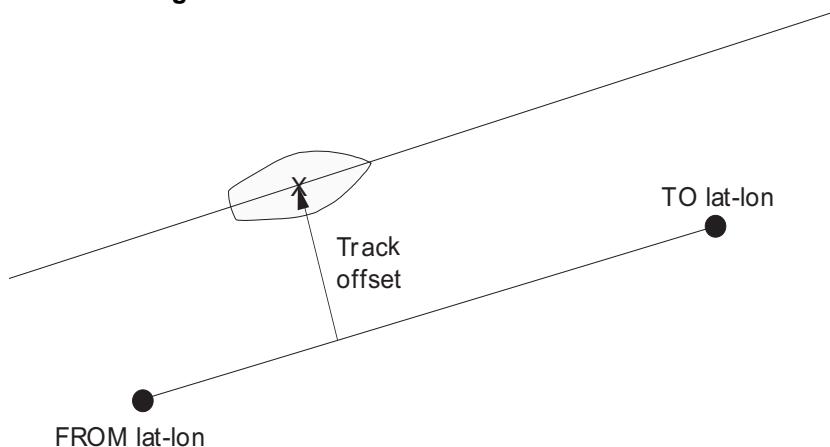
Factory Default:

setnav 90.0 0.0 90.0 0.0 0.0 from to

ASCII Example:

setnav 51.1516 -114.16263 51.16263 -114.1516 -125.23 from to

Figure 9: Illustration of SETNAV Parameters

> Consider the case of setting waypoints in a deformation survey along a dam. The surveyor enters the From and To point locations, on either side of the dam using the SETNAV command. They then use the NAVIGATE log messages to record progress and show where they are in relation to the From and To points.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETNAV 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	fromlat	± 90 degrees		Origin latitude in units of degrees/decimal degrees. A negative sign for South latitude. No sign for North latitude	Double	8	H
3	fromlon	± 180 degrees		Origin longitude in units of degrees/decimal degrees. A negative sign for West longitude. No sign for East longitude	Double	8	H+8
4	tolat	± 90 degrees		Destination latitude in units of degrees/decimal degrees	Double	8	H+16
5	tolon	± 180 degrees		Destination longitude in units of degrees/decimal degrees	Double	8	H+24
6	trackoffset	± 1000 km		Waypoint great circle line offset (in metres) establishes offset track. Positive indicates right of great circle line and negative indicates left of great circle line	Double	8	H+32

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
7	from-point	5 characters maximum		ASCII origin station name	String [max 5]	Variable[a]	H+40
8	to-point	5 characters maximum		ASCII destination station name	String [max 5]	Variable[a]	Variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.139 SETPREFERREDNETIF
Set the Network Interfaces for DNS and Default Gateway Configuration

OEM Platform: ProPak6

This command sets the preferred network interface. The preferred interface is used to obtain the default gateway and DNS server network configuration for the receiver.

The commands to configure the gateway and DNS server settings apply to specific interfaces. IPCONFIG and DNSCONFIG configure the Ethernet interface (ETHA), WIFICLICONFIG configures the Wi-Fi interface, and CELLULARCONFIG configures the cellular modem interface (CELL).

The gateway and DNS configuration set for the preferred interface can be static or DHCP. If an interface is configured to use DHCP, then the receiver will use the default gateway and DNS server received from DHCP only when received on the preferred interface.

Abbreviated ASCII Syntax: **Message ID: 1688**

SETPREFERREDNETIF [netifpreferredinterfacemode] primarynetif [secondarynetif] [tertiarynetif] [quaternarynetif] [quinarynetif] [senarynetif] [septenarynetif] [octonarynetif] [nonarynetif] [denarynetif]

> (i) When entering the SETPREFERREDNETIF command in ASCII, only the *PrimaryNetIf* field is required. See the ASCII example below.
>
> When entering the SETPREFERREDNETIF command in binary, all of the fields must be entered. However, all fields other than *PrimaryNetIf* must be set to zero.

Factory Default:

setpreferrednetif etha

ASCII Example:

setpreferrednetif cell

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	SETPREFERRED NETIF 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	NetIfPreferred InterfaceMode	Not required	0	Preferred interface selection behavior	Enum	4	H

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
3	PrimaryNetIf	ETHA	2	Set ETHA as the primary network interface	Enum	4	H+4
		WIFI	10	Set Wi-Fi as the primary network interface			
		CELL	20	Set the cellular modem as the primary network interface			
4	SecondaryNetIf	Not required	0	Secondary network interface	Enum	4	H+8
5	TertiaryNetIf	Not required	0	Tertiary network interface	Enum	4	H+12
6	QuaternaryNetIf	Not required	0	Quaternary network interface	Enum	4	H+16
7	QuinaryNetIf	Not required	0	Quinary network interface	Enum	4	H+20
8	SenaryNetIf	Not required	0	Senary network interface	Enum	4	H+24
9	SeptenaryNetIf]	Not required	0	Septenary network interface	Enum	4	H+28
10	OctonaryNetIf	Not required	0	Octonary network interface	Enum	4	H+32
11	NonaryNetIf	Not required	0	Nonary network interface	Enum	4	H+36
12	DenaryNetIf	Not required	0	Denary network interface	Enum	4	H+40

2.4.140 SETROVERID
Set ID for ALIGN rovers

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets the Rover ID output in ROVERPOS, HEADING2, ALIGNBSLNXYZ and ALIGNBSLNENU logs.

The default value for the ID is set using the last six characters of the receiver PSN Number. For example, if the receiver PSN number is DAB07170027, ID is set as R027, i.e., 17 is represented as R and last three characters are filled in as is. The fourth last character is ignored.

> It is not guaranteed that each receiver will have a unique auto-generated ID. Use this command to set the ID in case the auto-generated ID overlaps with other rovers. It is the user's responsibility to ensure each receiver ID is unique (if they own multiple receivers). If the ID overlaps, use this command to set the ID.

Abbreviated ASCII Syntax: **Message ID: 1135**

SETROVERID rov1id

Factory Default:

if the receiver psn is: dab07170027
setroverid r027

Input Example

setroverid rov1

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETROVERID 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	ID	4 Character String e.g., ROV1		ID String (maximum 4 characters plus NULL)	String[5]	5[a]	H

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.141 SETRTCM16
Enters ASCII text for RTCM data stream

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The RTCM type 16 message allows ASCII text to be transferred from a GNSS base station to rover GNSS receivers. The SETRTCM16 command is used to define the ASCII text at the base station. The text defined by the SETRTCM16 command can be verified in the RXCONFIG log. Once the ASCII text is defined it can be broadcast periodically by the base station with the command "log <port> RTCM16 ONTIME <interval>". The received ASCII text can be displayed at the rover by logging RTCM16T (refer to the logs under *Section 3.2.129* on *page 636*).

This command limits the input message length to a maximum of 90 ASCII characters. If the message string contains any delimiters (that is, spaces, commas, tabs and so on) the entire string must be contained in double quotation marks.

Abbreviated ASCII Syntax: **Message ID: 131**

SETRTCM16 text

Input Example:

setrtcm16 "base station will shut down in 1 hour"

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETRTCM16 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	text	Maximum 90 character string		The text string	String [max 90]	Variable[a]	H

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.142 SETRTCM36
Enters ASCII text with Russian characters

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The RTCM Type 36 message is the GLONASS equivalent of the RTCM Type 16 message except that the RTCM36 message can contain characters from an extended character set including Russian characters. *Table 64, System Used for Timing* on *page 308* provides the standard decimal and hex codes to use when transmitting Cyrillic characters to provide Russian language messages. Codes from 0 to 127 correspond to standard ASCII codes.

To support the 8-bit character data in the ASCII version, 8-bit characters are represented as \xnn (or \dnnn) which are the hexadecimal (or decimal) values of the characters. A "\" is represented as "\\".

In the RTCM36T log, the ASCII output displays the 8-bit characters in the decimal \dnnn representation. However, in the SETRTCM36 command, you can enter the 8-bit characters using the \x or \d prefix.

> ⓘ This command limits the input message length to a maximum of 90 ASCII characters. If the message string contains any delimiters (that is, spaces, commas, tabs, and so on) the entire string must be contained in double quotation marks.

Abbreviated ASCII Syntax: **Message ID: 880**

SETRTCM36 extdtext

Input Example:

To set the message "QUICK ШТОРМ", enter any of the following commands (color added to aid understanding):

        setrtcm36 "quick \d166\d146\d174\d144\d140"

        setrtcm36 "quick \xa6\x92\xae\x90\x8c"

        setrtcm36 "\x51\x55\x49\x43\x4b\x20\xa6\x92\xae\x90\x8c"

        setrtcm36 "\x51\x55\x49\x43\x4b\xa6\x92\xae\x90\x8c"

Similarly, the corresponding RTCM36T message, see *page 636*, looks like:

#RTCM36TA,COM1,0,77.5,FINESTEERING,1399,237244.454,00000000, 2E54,35359;"QUICK \D166\D146\D174\D144\D140"*4AA7F340

> ⓘ Similar to the RTCM Type 16 message, the SETRTCM36 command is used to define the ASCII text at the base station and can be verified in the RXCONFIG log. Once the ASCII text is defined it can be broadcast periodically by the base station with the command, for example "log< port> RTCM36 ONTIME 10". The received ASCII text can be displayed at the rover by logging RTCM36T.

Table 63: Russian Alphabet Characters (Ch) in Decimal (Dec) and Hexadecimal (Hex)

Hex Code	Dec Code	Ch	Hex Code	Dec Code	Ch	Hex Code	Dec Code	Ch	Hex Code	Dec Code	Ch
80	128	А	90	144	Р	A0	160	а	B0	176	р
81	129	Б	91	145	С	A1	161	б	B1	177	с
82	130	В	92	146	Т	A2	162	в	B2	178	т
83	131	Г	93	147	У	A3	163	г	B3	179	у
84	132	Д	94	148	Ф	A4	164	д	B4	180	ф
85	133	Е	95	149	Х	A5	165	е	B5	181	х
86	134	Ж	96	150	Ц	A6	166	ж	B6	182	ц
87	135	З	97	151	Ч	A7	167	з	B7	183	ч
88	136	И	98	152	Ш	A8	168	и	B8	184	ш
89	137	Й	99	153	Щ	A9	169	й	B9	185	щ
8A	138	К	9A	154	Ъ	AA	170	к	BA	186	ъ
8B	139	Л	9B	155	Ы	AB	171	л	BB	187	ы
8C	140	М	9C	156	Ь	AC	172	м	BC	188	ь
8D	141	Н	9D	157	Э	AD	173	н	BD	189	э
8E	142	О	9E	158	Ю	AE	174	о	BE	190	ю
8F	143	П	9F	159	Я	AF	175	п	BF	191	я

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETRTCM36 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	extdtext	Maximum 90 character string		The RTCM36 text string	String [max 90]	Variable[a]	H

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.143 SETRTCMRXVERSION
Sets the RTCM message standard

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets RTCM corrections to RTCM 2.2 or 2.3 message standards.

> (i) For RTCM correction message types, see *Table 37, Page 186*.

Abbreviated ASCII Syntax: **Message ID: 1216**

SETRTCMRXVERSION version

Factory Default:

setrtcmrxversion v23

Input Example:

setrtcmrxversion v23

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETRTCMRXVERSION 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	version	v23	0	RTCM version 2.3	Enum	4	H
		v22	1	RTCM version 2.2			

2.4.144 SETRTCMTXVERSION
Sets the RTCM transmission standard

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets RTCM corrections to RTCM 2.2 or 2.3 transmission standards.

> For RTCM correction message types, see *Table 37, Page 186*

Abbreviated ASCII Syntax: **Message ID: 1322**

SETRTCMTXVERSION version

Factory Default:

setrtcmtxversion v23

Input Example:

setrtcmtxversion v23

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETRTCMTXVERSION 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	version	v23	0	RTCM version 2.3	Enum	4	0
		v22	1	RTCM version 2.2			

2.4.145 SETTIMEBASE
Sets primary and backup systems for time base

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command configures the primary and backup steering system(s) for timing. The primary system is the system that the receiver steers the clock to. Upon startup, the primary system must be present long enough to steer the clock to be valid once, otherwise, the backup system cannot be used. The backup system is used whenever the primary system is not present.

Abbreviated ASCII Syntax: **Message ID: 1237**

settimebase primarysystem numbackups[system[timeout]]

Factory Default:

For GLONASS only receiver:

       settimebase glonass 0

For GPS capable receiver:

       settimebase gps 1 auto 0

For BeiDou only receiver:

       settimebase beidou 0

Input Example:

       settimebase gps 1 glonass 30

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETTIMEBASE 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	primarysystem	*Table 64, Page 308*		The primary system for steering the receiver clock	Enum	4	H
3	numbackups	0, 1, 2 or 3		The number of records to follow	Ulong	4	H+4
4	system[a]	*Table 64, Page 308*		The system to be used for backup	Enum	4	H+8
5	timeout[a]	0 to +4294967295 (seconds)		Duration that the backup system is used to steer the clock. 0 means ongoing	Ulong	4	H+12

a. Fields can repeat.

Table 64: System Used for Timing

Binary	ASCII
0	GPS
1	GLONASS
2	GALILEO
3	BEIDOU
99	AUTO[a]

a. AUTO is used only as a backup system (not available for primary system field).

2.4.146 SETTROPOMODEL
Sets Troposphere model

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets the troposphere model used to correct ranges used in the PSRPOS and PDPPOS solutions.

Abbreviate ASCII Syntax: **Message ID: 1434**

SETTROPOMODEL model

Factory Default:

settropomodel auto

Input Example:

settropomodel none

> Disabling the troposphere model may negatively affect positioning accuracy. NovAtel recommends that only advanced users modify this setting.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETTROPOMODEL 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	model	NONE	1	Do not apply any troposphere corrections	Enum	4	H
		AUTO	2	Automatically use an appropriate model			

2.4.147 SETUTCLEAPSECONDS
Sets future leap seconds

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command allows the user to force the UTC offset to be updated according to the input date.

Abbreviated ASCII Syntax: **Message ID: 1150**

SETUTCLEAPSECONDS seconds [futureweeknumber [futuredaynumber [futureseconds]]]

Factory Default:

       setutcleapseconds 16 1694 7 16

Input Example:

       setutcleapseconds 17 1823 7 16

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SETUTCLEAPSECONDS 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	Seconds[a]	0-		Current UTC leap second	Ulong	4	H
3	Futureweeknumber	0-10000		GPS Week when future leap seconds will take effect (default = 1694)	Ulong	4	H+4
4	Futuredaynumber	1-7		Day of the week when future leap seconds will take effect (default = 7)	Ulong	4	H+8
5	Futureseconds	0-		Future leap second offset that will take effect at the end of the futuredaynumber of the futureweeknumber (default = 16)	Ulong	4	H+12

a. This value is only be applied if the UTC status in the TIME log is not Valid.

2.4.148 SOFTLOADCOMMIT
Completes the SoftLoad process

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command completes the SoftLoad process by verifying the downloaded image and activating it. Refer to The OEM6 Installation and Operation Manual (OM-20000128) for more information about the SoftLoad process.

This command can only be sent to the receiver when the SOFTLOADSTATUS log reports READY_FOR_DATA.

After issuing the SOFTLOADCOMMIT command the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 300 seconds from the time the command was received by the receiver. If an error response is returned, consult the SOFTLOADSTATUS log for more detail.

Abbreviated ASCII Syntax: **Message ID: 475**

SOFTLOADCOMMIT

Input Example:

softloadcommit

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SOFTLOADCOMMIT 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	Reserved	-		Reserved. Set to 1 in the binary case	Enum	4	H

2.4.149 SOFTLOADDATA
Sends firmware image data to the receiver for the SoftLoad process

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is only valid in binary mode.

This command is used to upload data to the receiver for the SoftLoad process. Refer to the OEM6 Installation and Operation Manual (OM-20000128) for more information about the SoftLoad process.

After each SOFTLOADDATA command, the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 15 seconds from the time the command was received by the receiver. If an error response is returned, consult the SOFTLOADSTATUS log for more detail.

This command can only be sent to the receiver once the SOFTLOADSREC or SOFTLOADSETUP commands have sent the content of the S0 records from the start of a firmware *.hex or *.shex file. In these cases, the SOFTLOADSTATUS log reports READY_FOR_SETUP or READY_FOR_DATA.

Abbreviated ASCII Syntax: **Message ID: 1218**

not applicable

Field	Field Type	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SOFTLOADDATA header	-	NovAtel binary message header	-	H	0
2	offset	-	Offset of the data within the downloaded image	Ulong	4	H
3	data length	-	Number of bytes of data. This must match the number of bytes contained within the "data" field	Ulong	4	H+4
4	data	-	Incoming data up to a maximum of 4096 bytes	Uchar	4096	H+8

2.4.150 SOFTLOADFILE

Updates the receiver firmware using a *.hex or *.shex file that has been uploaded to the receiver

OEM Platform: 638, ProPak6

Use this command to update receiver firmware using the file devices of the receiver.

> (i) If the Storage device is USBSTICK, copy the firmware .hex file to the USB stick before connecting the stick to the OEM638.
>
> If the Storage device is the INTERNALFLASH, use the FTP server to upload the firmware .hex file to the internal flash (eMMC) data logging device.

Abbreviated ASCII Syntax: **Message ID: 1302**

SOFTLOADFILE storage file [dest]

Abbreviated ASCII Example:

softloadfile usbstick zm6omap.hex

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	SOFTLOAD FILE 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	storage	USBSTICK	1	The source device the on which the file is stored.	Enum	4	H
		INTERNAL_FLASH	4				
3	file	String		The hex file to load. File names have a 128 character limit	Char [128]	128	H+4
4	dest	Reserved	0	Reserved	Enum	4	variable

2.4.151 SOFTLOADRESET
Initiates a new SoftLoad process

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command restarts the SoftLoad process. Refer to the OEM6 Installation and Operation Manual (OM-2000128) for more information about the SoftLoad process.

The command does not affect the flash and does not reset the receiver.

The SOFTLOADRESET command can be issued at any time. If it is issued while a SoftLoad process is currently in progress then that process is terminated and a new one is started. After the SOFTLOADRESET command is processed the SOFTLOADSTATUS log will report a status of READY_FOR_SETUP.

After issuing the SOFTLOADRESET command the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 300 seconds from the time the command was received by the receiver. If an error response is returned, consult the SOFTLOADSTATUS log for more detail.

Abbreviated ASCII Syntax: **Message ID: 476**

SOFTLOADRESET

Input Example:

    softloadreset

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SOFTLOADRESET 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	Reserved	-		Reserved. Set to 1 in the binary case	Enum	4	H

2.4.152 SOFTLOADSETUP
Sends configuration information to the receiver for the SoftLoad process

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The SOFTLOADSETUP command can be used in place of the SOFTLOADSREC command when sending S0 Records. This command is meant to be used if the user requires that the entire SoftLoad process be performed in binary, but can also be used in ASCII or abbreviated ASCII. The examples below are given in abbreviated ASCII for simplicity.

Refer to the OEM6 Installation and Operation Manual (OM-20000128) for more information about the SoftLoad process.

This command can only be sent to the receiver when the SOFTLOADSTATUS log reports READY_FOR_SETUP.

After each SOFTLOADSETUP command, the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 15 seconds from the time the command was received by the receiver. If an error response is returned, consult the SOFTLOADSTATUS log for more detail.

NovAtel S0 records use the following format: S0~X~<<DATA>>, where X is the Setup Type and <<DATA>> is a NULL terminated string. To convert from S0 record to the SOFTLOADSETUP command, convert the Setup Type to the appropriate Setup type enumeration, as described in Table 64, and copy the <<DATA>> string in to the Setup data string.

Abbreviated ASCII Syntax: **Message ID: 1219**

SOFTLOADSETUP setuptype setupdata

Input Example:

        softloadsetup datatype "APP"

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SOFTLOADSETUP 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	Setup type	See *Table 65, Page 316*		The type of setup command	Enum	4	H

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
3	Setup data	-	-	ASCII setup data string. See Table 65 for details on this data. This data can be pulled from the S0 records of the hex file being loaded onto the receiver. If the ASCII form of this command is used, this string must be enclosed in double quotes (" ")	String [512]	variable[a]	H+4

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 65: Available Set Up Commands

Binary	ASCII	Description
1	Platform	Comma separated list of platforms supported by the data to be uploaded. This corresponds to S0~P~. For example, the S-Record S0~P~OEM628,OEM628R,OEM615, translates to SOFTLOADSETUP PLATFORM "OEM628,OEM628R,OEM615"
2	Version	Version of the data to be uploaded. This corresponds to S0~V~. For example, the S-Record S0~V~OMP060400RN0000, translates to SOFTLOADSETUP VERSION "OMP060400RN0000"
3	Datatype	Intended data block for the data to be uploaded. This corresponds to S0~T~. For example, the S-Record S0~T~APP, translates to SOFTLOADSETUP DATATYPE "APP"
4	Authcode	PSN and AUTH code for the data to be uploaded. The format is: PSN:AuthCode.Note that since there are commas within the AuthCode, double quotes must surround the PSN:AuthCode string. For example: SOFTLOADSETUP AUTHCODE "BFN10260115:T48JF2,W25DBM,JH46BJ,2WGHMJ,8JW5TW,G2SR0RCCR,101114"

2.4.153 SOFTLOADSREC
Sends an S-Record to the receiver for the SoftLoad process

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use this command to send S-Records to the receiver for the SoftLoad process. Refer to the [OEM6 Installation and Operation Manual](#) (OM-20000128) for more information about the SoftLoad process.

After each `SOFTLOADDATA` command, the user must wait for the OK or ERROR command response before proceeding. This response is guaranteed to be output from the receiver within 15 seconds from the time the command was received by the receiver. If an error response is returned, consult the SOFTLOADSTATUS log for more detail.

This command can only be sent to the receiver when the SOFTLOADSTATUS log reports READY_FOR_SETUP or READY_FOR_DATA.

Abbreviated ASCII Syntax: **Message ID: 477**

SOFTLOADSREC s-record

Input Example:

       softloadsrec "S30900283C10FAA9F000EF"

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SOFTLOADSREC 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	SREC	-		ASCII S-Record string copites from firmware *.hex or *.shex file	String [515]	variable[a]	H
3	Reserved	-	1	Reserved. Set to 1 in the binary case	Ulong	4	variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

2.4.154 SOFTPOWER
Shut down the receiver

OEM Platform: ProPak6

Use the SOFTPOWER command to power off the receiver.

Before the receiver is shut down, the internal flash storage is unmounted to minimize the risk of losing data being recorded to a log file.

Abbreviated ASCII Syntax: **Message ID: 213**

SOFTPOWER pdc_priorityenum

Input Example:

softpower now

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SOFTPOWER 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	PDC_PriorityEnum	NOW	1	Power off immediately	Enum	4	H

2.4.155 STATUSCONFIG
Configures RXSTATUSEVENT mask fields

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to configure the various status mask fields in the RXSTATUSEVENT log, see *page 668*. These masks can modify whether various status fields generate errors or event messages when they are set or cleared.

Receiver Errors automatically generate event messages. These event messages are output in RXSTATUSEVENT logs. It is also possible to have status conditions trigger event messages to be generated by the receiver. This is done by setting/clearing the appropriate bits in the event set/clear masks. The set mask tells the receiver to generate an event message when the bit becomes set. Likewise, the clear mask causes messages to be generated when a bit is cleared. To disable all these messages without changing the bits, simply UNLOG the RXSTATUSEVENT logs on the appropriate ports. Refer also to the *Built in Status Tests* chapter in The OEM6 Installation and Operation Manual (OM-20000128).

Abbreviated ASCII Syntax: **Message ID: 95**

STATUSCONFIG type word mask

Factory Default:

STATUSCONFIG PRIORITY STATUS 0
STATUSCONFIG PRIORITY AUX1 0x00000008
STATUSCONFIG PRIORITY AUX2 0
STATUSCONFIG SET STATUS 0x00000000
STATUSCONFIG SET AUX1 0
STATUSCONFIG SET AUX2 0
STATUSCONFIG CLEAR STATUS 0x00000000
STATUSCONFIG CLEAR AUX1 0
STATUSCONFIG CLEAR AUX2 0

ASCII Example:

statusconfig set status 0028a51d

ⓘ The receiver gives the user the ability to determine the importance of the status bits. In the case of the Receiver Status, setting a bit in the priority mask causes the condition to trigger an error. This causes the receiver to idle all channels, set the ERROR strobe line, flash an error code on the status LED, turn off the antenna (LNA power) and disable the RF hardware, the same as if a bit in the Receiver Error word is set. Setting a bit in an Auxiliary Status priority mask causes that condition to set the bit in the Receiver Status word corresponding to that Auxiliary Status.

Table 66: Mask Types

ASCII	Binary	Description
PRIORITY	0	Replace the Priority mask
SET	1	Replace the Set mask
CLEAR	2	Replace the Clear mask

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	STATUSCONFIG 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	type	See *Table 66, Page 320*		Type of mask to replace	Enum	4	H
3	word	STATUS	1	Receiver Status word	Enum	4	H+4
		AUX1	2	Auxiliary 1 Status word			
		AUX2	3	Auxiliary 2 Status word			
		AUX3	4	Auxiliary 3 Status word			
4	mask	8 digit hexadecimal		The hexadecimal bit mask	Ulong	4	H+8

2.4.156 *STEADYLINE*
Configures position mode matching

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The STEADYLINE functionality helps mitigate the discontinuities that often occur when a GNSS receiver changes positioning modes. The effect is especially evident when a receiver transitions from an RTK position mode solution to a lower accuracy "fall back" solution, such as Omnistar HP/XP, DGPS, WAAS+GLIDE or even autonomous GLIDE. Smooth transitions are particularly important for agricultural steering applications where sudden jumps may be problematic.

The STEADYLINE internally monitors the position offsets between all the positioning modes present in the receiver. When the receiver experiences a position transition, the corresponding offset is applied to the output position to limit a potential real position jump. When the original accurate position type returns, the STEADYLINE algorithm will slowly transition back to the new accurate position at a default rate of 0.005 m/s. This creates a smoother pass-to-pass relative accuracy at the expense of a possible degradation of absolute accuracy.

For example, a receiver can be configured to do both RTK and GLIDE. If this receiver has a fixed RTK position and experiences a loss of correction data causing the loss of the RTK solution it will immediately apply the offset between the two position modes and uses the GLIDE position stability to maintain the previous trajectory. Over time the GLIDE (or non-RTK) position will experience some drift. Once the RTK position is achieved again the receiver will start using the RTK positions for position stability and will slowly transition back to the RTK positions at a default rate of 0.005 m/s.

If the position type is OUT_OF_BOUNDS (see UALCONTROL *page 331*) then STEADYLINE is reset.

Abbreviated ASCII Syntax: **Message ID: 1452**

STEADYLINE mode [transition_time]

Factory Default:

steadyline disable

ASCII Example:

steadyline prefer_accuracy 100

Table 67: STEADYLINE Mode

ASCII	Binary	Description
DISABLE	0	Disable mode match
MAINTAIN	1	Maintain the relative offset of the solution. There is no discontinuity in the position solution when the reference position type changes. Any offset in the position is maintained.
TRANSITION	2	Transition, at a user-configurable rate. There is no discontinuity in the position solution when the reference position type changes. The position will slowly transition to the new reference position type over the time period specified by the Transition time parameter.
RESET	3	Reset the saved mode match offsets
PREFER_ACCURACY	4	TRANSITION when changing from less accurate reference positioning type to more accurate reference positioning type. MAINTAIN when changing from more accurate reference positioning type to a less accurate reference positioning type.
UAL	5	For use with the `UALCONTROL` command: TRANSITION when the position type is in WARNING MAINTAIN when the position type is in OPERATIONAL DISABLE when the position type is OUT_OF_BOUNDS

Field #	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	STEADYLINE 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	mode	Table 67 above		STEADYLINE mode	Enum	4	H
3	Transition time			Time over which solutions will transition in seconds. The minimum rate of change is 0.005 m/s regardless of this parameter.	Ulong	4	H+4

2.4.157 TERRASTARPAYG
Enables use of TerraStar Pay-As-You-Go time

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command enables use of TerraStar Pay-As-You-Go (PAYG) time. For TerraStar term subscriptions, the corrections are used automatically. For PAYG subscriptions, the use of the corrections must be explicitly enabled using this command. This guards against inadvertent use of the PAYG period. Furthermore, in conjunction with the ASSIGNLBANDBEAM command (*page 81*), this command permits the receiver to track and receive subscriptions from TerraStar L-Band satellites without using the purchased PAYG period.

Abbreviated ASCII Syntax: **Message ID: 1749**

TERRASTARPAYG switch

Factory Default:

terrastarpayg disable

ASCII Example:

terrastarpayg enable

Field #	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	TERRASTAR PAYG header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see *Section 1.1, Message Types on Page 21*)	-	H	0
2	switch	DISABLE	0	Enable use of PAYG corrections	Enum	4	H
		ENABLE	1	Disable use of PAYG corrections			

2.4.158 THISANTENNAPCO
Sets the PCO model of this receiver

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use the THISANTENNAPCO command to set the Phase Center Offsets (PCO) for the given frequency of this receiver. The Offsets are defined as North, East and Up from the Antenna Reference Point to the Frequency Phase Center in mm.

Abbreviated ASCII Syntax: **Message ID: 1417**

THISANTENNAPCO frequency[north offset][east offset][up offset]

ASCII Example:

thisantennapco gpsl1 0.61 1.99 65.64

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	THISANTENNA PCO 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	Frequency	*Table 16, Frequency Type* on *page 87*		The frequency for which the phase center offsets are valid.	Enum	4	H
3	North Offset			NGS standard Phase Center North Offset in millimetres.[a]	Double	8	H+4
4	East Offset			NGS standard Phase Center East Offset in millimetres.[a]	Double	8	H+12
5	Up Offset			NGS standard Phase Center Up Offset in millimetres.[a]	Double	8	H+20

a. Enter values as per the NGS standards and tables to define which direction is plus or minus.

2.4.159 THISANTENNAPCV
Sets the PCV model of this receiver

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use the THISANTENNAPCV command to set the Phase Center Variation (PVC) for the given frequency of this receiver. The Phase Center Variation entries follow the NGS standard and correspond to the phase elevation at 5 degree increments starting at 90 degrees and decreasing to 0.

Abbreviated ASCII Syntax: **Message ID: 1418**

THISANTENNAPCV frequency[pcv array]

ASCII Example:

> thisantennapcv gpsl1 0.00 -0.020 -0.07 -0.15 -0.24 -0.34 -0.43 -0.51 -0.56 -0.61 -0.65 -0.69 -0.69 -0.62 -0.44 -0.13 0.28 0.70 1.02

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	THISANTENNA PCV 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	Frequency	*Table 16, Frequency Type* on *page 87*		The frequency for which the phase center variations is valid.	Emun	4	H
3	PCV Array			NGS standard 19 Element array of Phase Center Variations for phase variation for 5 degree elevation increments starting at 90 degrees and decreasing to 0. The variances are entered in millimetres.	Double Array [19]	152	H+4

2.4.160 THISANTENNATYPE
Sets the antenna type of this receiver

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Use the THISANTENNATYPE command to set the antenna type of this receiver. The antenna type and radome type are the NGS names for the antenna.

> (i) When antenna type is set using this command, the receiver will look up and use the Phase Center Variations and Phase Center Offsets from an internal table.

Abbreviated ASCII Syntax: **Message ID: 1420**

THISANTENNATYPE antennatype[radometype]

ASCII Example:

thisantennatype nov702

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	THISANTENNA TYPE 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	antenna type	*Table 17, Antenna Type* on *page 90*	NGS Antenna Name	Enum	4	H
3	radome type	*Table 18, Radome Type* on *page 95*	NGS Radome Name	Enum	4	H+4

2.4.161 TRACKSV
Overrides automatic satellite assignment criteria

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to override the automatic satellite/channel assignment for all satellites with manual instructions.

Abbreviated ASCII Syntax: **Message ID: 1326**

TRACKSV system SVID condition

Factory Default:

> GPS, GLONASS, Galileo, QZSS and BeiDou default = GOODHEALTH
> SBAS default = ANYHEALTH

> (i) For the OEM617D and FlexPak6D receivers, this command applies to both the primary and secondary antennas.

Table 68: TrackSV Command Condition

Binary	ASCII	Description
1	NEVER	Never track this satellite
2	GOODHEALTH	Track this satellite if the health is indicated as healthy in both the almanac and ephemeris
3	ANYHEALTH	Track this satellite regardless of health status
4	ALWAYS	Always track this satellite

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	TRACKSV 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	System	see *Table 110, Satellite System* on *page 508*		System that the SVID belongs to	Enum	4	H
3	SVID	GPS: 1-32 SBAS: 120-138, 183-187 GLONASS:1-24 (see *1.3 GLONASS Slot and Frequency Numbers* on *page 34*) GALILEO: 1-36 QZSS: 193-197 BeiDou: 1-30 "0" allowed and applies to all SVIDs for the specified system type		Satellite SVID number	Ulong	4	H+4

Field	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
4	Condition	see *Table 68, TrackSV Command Condition* on *page 327*		Tracking condition	Enum	4	H+8

2.4.162 *TUNNELESCAPE*
Breaks out of an established tunnel

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The tunnel escape sequence feature allows you to break out of a tunnel between two ports by sending a predefined sequence of bytes through the tunnel in-line with the data stream. While the Bluetooth implementation on OEM628 and ProPak6 products utilizes the tunnel mode of OEM receivers, the tunnel escape sequence feature is applicable to any tunneling application.

Use the TUNNELESCAPE command to specify the tunnel escape sequence. The escape sequence is applied independently to all active tunnels. Use the SAVECONFIG command to save the escape sequence in case of a power cycle.

This command is used to define an escape sequence that, when detected in a byte stream between any two COM (or AUX) ports, resets the interface mode to NOVATEL NOVATEL on those ports. The baud rate and other port parameters remain unaffected.

The TUNNELESCAPE command accepts three parameters. The first is the *switch* parameter with ENABLE or DISABLE options. The second is the *length* parameter. It is a number from 1 to 8 and must be present if the *switch* parameter is set to ENABLE. The third parameter, *esc seq*, consists of a series of pairs of digits representing hexadecimal numbers, where the number of pairs are equal to the value entered for the second parameter. The series of hexadecimal pairs of digits represent the escape sequence. The receiver detects a sequence in a tunnel exactly as it was entered.

For example, the command TUNNELESCAPE ENABLE 4 61626364 searches for the bytes representing "abcd" in a tunnel stream. TUNNELESCAPE ENABLE 3 AA4412 searches for the NovAtel binary log sync bytes.

You must first set up a tunnel. For example, create a tunnel between COM1 and COM2 by entering INTERFACEMODE COM1 TCOM2 NONE OFF. The commands can be entered in any order.

> 1. All bytes, leading up to and including the escape sequence, pass through the tunnel before it is reset. Therefore, the escape sequence is the last sequence of bytes that passes through the tunnel. Configure the receiver to detect and interpret the escape sequence. For example, use this information to reset equipment or perform a shutdown process.
>
> 2. The receiver detects the escape sequence in all active tunnels in any direction.
>
> 3. Create tunnels using the INTERFACEMODE command, see *page 184*.

Abbreviated ASCII Syntax: **Message ID: 962**

TUNNELESCAPE switch length escseq

Factory Default:

> tunnelescape disable 0

ASCII Example:

> tunnelescape enable 1 aa

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	TUNNEL-ESCAPE 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	switch	DISABLE	0	Enable or disable the tunnel escape mode (default = DISABLE)	Enum	4	H
		ENABLE	1				
3	length	1 to 8		Specifies the number of hex bytes to follow	Ulong	4	H+4
4	escseq			Escape sequence where Hex pairs are entered without spaces, for example, AA4412	Uchar[8]	8	H+8

> ▼ If using the SAVECONFIG command in NovAtel Connect, ensure all windows other than the *Console* window are closed. If open, Connect also saves log commands used for its various windows. This results in unnecessary data being logged.

2.4.163 UALCONTROL
Setup User Accuracy levels

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The UALCONTROL command is used to define User Accuracy Levels. User accuracy levels are user defined standard deviations thresholds, used to determine solution acceptability. Issuing the UALCONTROL command causes the BESTPOS and GPGGA solution types to be controlled via the specified thresholds, rather than by the solution source or mode. The new solution types are described in the table below.

Table 69: User Accuracy Level Supplemental Position Types and NMEA Equivalents

Value	BESTPOS Position Type[a]	NMEA Equivalent[b]
70	OPERATIONAL	4
71	WARNING	5
72	OUT_OF_BOUNDS	1

a. As reported in the BESTPOS log (*page 406*).
b. Refers to the GPGGA quality indicator (see *page 474* for details).

The SETBESTPOSCRITERIA command (*page 291*) determines which standard deviations are compared against the provided thresholds. When using the STEADYLINE command (*page 321*) together with the UALCONTROL command (*page 331*), the UAL setting is recommended. Refer to *Table 67, STEADYLINE Mode* on *page 322* for mode details.

> ⓘ UAL is useful for applications that rely upon specific solutions types being present in the BESTPOS or GPGGA logs. For example, if an agricultural steering systems commonly require an RTK fixed GPGGA solution type (4) to operate, and interruptions in RTK conventionally cause the GPGGA to switch to another solution type. This causes the steering system to disengage. However, while using STEADYLINE, solutions with fixed RTK accuracy can be maintained by GLIDE even if RTK is interrupted. UALCONTROL can be used to ensure that the required solution type is maintained through such interruptions, permitting the steering system to function continuously.

Abbreviated ASCII Syntax: **Message ID: 1627**

UALCONTROL action [operational_limit][warning_limit]

Factory Default:

ualcontrol disable

ASCII Example:

ualcontrol enable 0.10 0.20

Table 70: UALCONTROL Action Parameters

ASCII	Binary	Description
DISABLE	0	Disables this feature
ENABLE	1	Replace BESTPOS and GPGGA position types with OPERATIONAL, WARNING or OUT_OF_BOUNDS based on the entered standard deviations (refer to *Table 69, User Accuracy Level Supplemental Position Types and NMEA Equivalents* on *page 331*)
CLEAR	2	Disable this feature and reset the entered standard deviations.

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	UALCONTROL header	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively (see *1.1 Message Types* on *page 21*)	-	H	0
2	Action	Commands action (see Table 70 above)	Enum	4	H
3	Operational Limit	Standard deviation in metres to report OPERATIONAL	Double	8	H+4
4	Warning Limit	Standard deviation in metres to report WARNING Note: OUT_OF_BOUND reports when the standard deviation exceeds this value	Double	8	H+12

2.4.164 UNASSIGN
Unassigns a previously assigned channel

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command cancels a previously issued `ASSIGN` command (*page 72*) and the SV channel reverts to automatic control (the same as ASSIGN AUTO).

Abbreviated ASCII Syntax: **Message ID: 29**

UNASSIGN channel [state]

Input Example:

> unassign 11

> (i) Issuing the `UNASSIGN` command to a channel that was not previously assigned by the `ASSIGN` command has no effect.

> **617D**
> **FPk6D**
> For the OEM617D and FlexPak6D receivers, when using the `UNASSIGN` command for SV channels on the primary antenna, the SV channel count goes from 0 to N-1, where N is the number of channels in the primary antenna channel configuration. When using the `UNASSIGN` command for SV channels on the secondary antenna, the SV channel count begins at N and goes to N+(M-1), where M is the number of SV channels in the secondary antenna channel configuration.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	UNASSIGN 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	channel	0 to n, where n is the number of the last channel in the current channel configuration		Channel number reset to automatic search and acquisition mode	Ulong	4	H
3	state	These return SV channel control to the automatic search engine immediately *(see Table 12, Page 73)*		Set the SV channel state (currently ignored)	Enum	4	H+4

2.4.165 UNASSIGNALL
Unassigns all previously assigned channels

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command cancels all previously issued ASSIGN commands for all SV channels (same as ASSIGNALL AUTO). Tracking and control for each SV channel reverts to automatic mode.

Abbreviated ASCII Syntax: **Message ID: 30**

UNASSIGNALL [system]

Input Example:

> unassignall gps

> (i) Issuing the UNASSIGNALL command has no effect on channels that were not previously assigned using the ASSIGN command.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	UNASSIGNALL 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	system	See *Table 13, Channel System* on *page 75*		System that will be affected by the unassignall command (default = ALL)	Enum	4	H

2.4.166 UNDULATION
Chooses undulation

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command permits you to either enter a specific geoidal undulation value. In the option field, the EGM96 table provides ellipsoid heights at a 0.5° by 0.5° spacing while the OSU89B is implemented at a 2° by 3° spacing. In areas of rapidly changing elevation, you could be operating somewhere within the 2° by 3° grid with an erroneous height. EGM96 provides a more accurate model of the ellipsoid which results in a denser grid of heights. It is also more accurate because the accuracy of the grid points themselves has also improved from OSU89B to EGM96. For example, the default grid (EGM96) is useful where there are underwater canyons, steep drop-offs or mountains.

The undulation values reported in the position logs are in reference to the ellipsoid of the chosen datum.

Abbreviated ASCII Syntax: **Message ID: 214**

UNDULATION option [separation]

Factory Default:

undulation egm96 0.0000

ASCII Example 1:
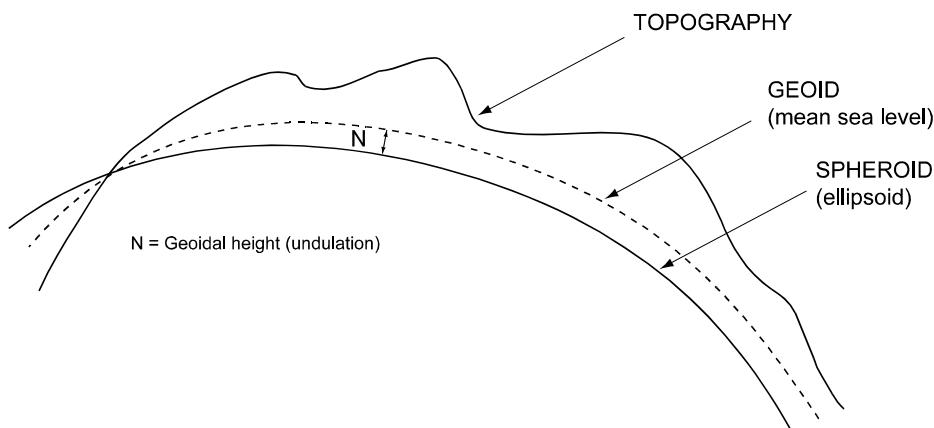
undulation osu89b

ASCII Example 2:

undulation user -5.599999905

Refer to the application note APN-006 Geoid Issue, available on our website www.novatel.com/support/search/ for a description of the relationships in *Figure 9*.

Figure 10: Illustration of Undulation

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	UNDULATION 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	option	USER	1	Use the user specified undulation value	Enum	4	H
		OSU89B	2	Use the OSU89B undulation table			
		EGM96	3	Use global geoidal height model EGM96 table			
3	separation	± 1000.0 m		The undulation value (required for the USER option) (default = 0.000)	Float	4	H+4

2.4.167 UNLOCKOUT
Reinstates a satellite in the solution

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command allows a satellite which has been previously locked out (LOCKOUT command, *page 200*) to be reinstated in the solution computation. If more than one satellite is to be reinstated, this command must be reissued for each satellite reinstatement.

Abbreviated ASCII Syntax: **Message ID: 138**

UNLOCKOUT prn

Input Example:

> unlockout 8

> (i) The UNLOCKOUT command is used to reinstate a satellite while leaving other locked out satellites unchanged.
>
> This command can be used for GPS, GLONASS, SBAS and QZSS.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	UNLOCKOUT 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	prn	GPS: 1-32 SBAS: 120-138, 183-187 GLONASS: see *1.3 GLONASS Slot and Frequency Numbers* on *page 34* QZSS 193-197	A single satellite PRN number to be reinstated	Ulong	4	H

2.4.168 UNLOCKOUTALL
Reinstates all previously locked out satellites

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command allows <u>all</u> satellites which have been previously locked out (LOCKOUT (*page 200*) or LOCKOUTSYSTEM (*page 201*) commands) to be reinstated in the solution computation.

> ▼ This command cannot be used in conjunction with SAVECONFIG to automatically remove the factory default LOCKOUTSYSTEM. It must be issued each time the receiver is started up.

Abbreviated ASCII Syntax: **Message ID: 139**

UNLOCKOUTALL

Input Example:

    unlockoutall

2.4.169 UNLOCKOUTSYSTEM
Reinstates previously locked out system

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command allows a system which has been previously locked out (refer to the LOCKOUTSYSTEM command, *page 201*) to be reinstated in the solution computation.

> (i) If more than one system is to be reinstated, this command must be reissued for each system reinstatement.

> ▼ This command cannot be used in conjunction with SAVECONFIG to automatically remove the factory default LOCKOUTSYSTEM. It must be issued each time the receiver is started up.

Abbreviated ASCII Syntax: **Message ID: 908**

UNLOCKOUTSYSTEM system

Input Example:

        unlockoutsystem glonass

> (i) The UNLOCKOUTSYSTEM command is used to reinstate a system while leaving other locked out systems unchanged.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	UNLOCKOUT 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	system	See *Table 110, Satellite System* on *page 508*		A single satellite system to be reinstated	Enum	4	H

2.4.170 UNLOG
Removes a log from logging control

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command is used to remove a specific log request from the system.

The [*port*] parameter is optional. If [*port*] is not specified, it is defaulted to the port on which the command was received.

Abbreviated ASCII Syntax: **Message ID: 36**

UNLOG [port] message

Input Example:

> unlog com1 bestposa
> unlog bestposa

(i) The UNLOG command is used to remove one or more logs while leaving other logs unchanged.

Field	Field Name	Binary Value	Description	Field Type	Binary Bytes	Binary Offset
1	UNLOG (binary) header	(See *Table 3, Page 26*)	This field contains the message header	-	H	0
2	port	See *Table 4, Page 27* (decimal port values greater than 16 may be used)	Port to which log is being sent (default = THISPORT)	Enum	4	H
3	message	Any valid message ID	Message ID of log to output (refer to *Table 81, OEM6 Logs in Numerical Order* on *page 375*) for a list of message ID numbers	Ushort	4	H+4
4	message type	Bits 0-4 = Reserved Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response Bit (*1.2 Responses* on *page 32*) 0 = Original Message 1 = Response Message	Message type of log	Char	1	H+6
5	Reserved			Char	1	H+7

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	UNLOG (ASCII) 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	See *Table 4, Page 27* (decimal port values greater than 16 may be used)		Port to which log is being sent (default = THISPORT)	Enum	4	H
3	message	Message Name	N/A	Message Name of log to be disabled *Table 10, OEM6 Commands in Alphabetical Order* on *page 48*	Ulong	4	H+4

2.4.171 UNLOGALL
Removes all logs from logging control

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

If [*port*] is specified, this command disables all logs on the specified port only. All other ports are unaffected. If [*port*] is not specified this command defaults to the ALL_PORTS setting.

Abbreviated ASCII Syntax: **Message ID: 38**

UNLOGALL [port] [held]

Input Example:

> unlogall com2_15
> unlogall true

ⓘ The `UNLOGALL` command is used to remove all log requests currently in use.

.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	UNLOGALL 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	See *Table 4, Page 27* (decimal values greater than 16 may be used)		Port to clear (default = ALL_PORTS)	Enum	4	H
3	held	FALSE	0	Does not remove logs with the HOLD parameter (default)	Bool	4	H+4
		TRUE	1	Removes previously held logs, even those with the HOLD parameter			

2.4.172 USERDATUM
Sets user customized datum

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command permits entry of customized ellipsoidal datum parameters. This command is used in conjunction with the DATUM command, see *page 118*. If used, the command default setting for USERDATUM is WGS84.

When the USERDATUM command is entered, the USEREXPDATUM command is then issued internally with the USERDATUM command values (*page 345*). It is the USEREXPDATUM command that appears in the RXCONFIG log. If the USEREXPDATUM or the USERDATUM command are used, their newest values overwrite the internal USEREXPDATUM values.

The transformation for the WGS84 to Local used in the OEM6 family is the Bursa-Wolf transformation or reverse Helmert transformation. In the Helmert transformation, the rotation of a point is counter clockwise around the axes. In the Bursa-Wolf transformation, the rotation of a point is clockwise. Therefore, the reverse Helmert transformation is the same as the Bursa-Wolf.

Abbreviated ASCII Syntax: **Message ID: 78**

USERDATUM semimajor flattening dx dy dz rx ry rz scale

Factory Default:

        userdatum 6378137.0 298.2572235628 0.0 0.0 0.0 0.0 0.0 0.0 0.0

ASCII Example:

        userdatum 6378206.400 294.97869820000 -12.0000 147.0000 192.0000 0.0000 0.0000 0.0000 0.000000000

> ⓘ   Use the USERDATUM command in a survey to fix the position with values from another known datum so that the GNSS calculated positions are reported in the known datum rather than WGS84.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	USERDATUM 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	semimajor	6300000.0 - 6400000.0 m		Datum Semi-major Axis (a) in metres	Double	8	H
3	flattening	290.0 - 305.0		Reciprocal Flattening, $1/f = a/(a-b)$	Double	8	H+8

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
4	dx	± 2000.0		Datum offsets from local to WGS84. These are the translation values between the user datum and WGS84 (internal reference)	Double	8	H+16
5	dy	± 2000.0			Double	8	H+24
6	dz	± 2000.0			Double	8	H+32
7	rx	± 10.0 radians		Datum rotation angle about X, Y and Z. These values are the rotation from your local datum to WGS84. A positive sign is for counter clockwise rotation and a negative sign is for clockwise rotation	Double	8	H+40
8	ry	± 10.0 radians			Double	8	H+48
9	rz	± 10.0 radians			Double	8	H+56
10	scale	± 10.0 ppm		Scale value is the difference in ppm between the user datum and WGS84	Double	8	H+64

2.4.173 USEREXPDATUM
Sets custom expanded datum

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Like the USERDATUM command, this command allows you to enter customized ellipsoidal datum parameters. However, USEREXPDATUM literally means user expanded datum allowing you to enter additional datum information such as velocity offsets and time constraints. The 7 expanded parameters are rates of change of the initial 7 parameters. These rates of change affect the initial 7 parameters over time relative to the Reference Date provided by the user.

This command is used in conjunction with the DATUM command, see *page 118*. If this command is used without specifying any parameters, the command defaults to WGS84. If a USERDATUM command is entered, the USEREXPDATUM command is then issued internally with the USERDATUM command values (*page 343*). It is the USEREXPDATUM command that appears in the RXCONFIG log. If the USEREXPDATUM or the USERDATUM command are used, their newest values overwrite the internal USEREXPDATUM values.

Abbreviated ASCII Syntax: **Message ID: 783**

USEREXPDATUM semimajor flattening dx dy dz rx ry rz scale xvel yvel zvel xrvel yrvel zrvel scalev refdate

Factory Default:

> userexpdatum 6378137.0 298.25722356280 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
> 0.0 0.0 0.0

ASCII Example:

> userexpdatum 6378137.000 298.25722356280 0.000000000 0.000000000 0.000000000
> 0.00000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
> 0.000000000 0.0000 0.000000000 0.000000000 0.000000000 0.000000000

> ⓘ Use the USEREXPDATUM command in a survey to fix the position with values from another known datum so that the GPS calculated positions are reported in the known datum rather than WGS84. For example, it is useful for places like Australia, where the continent is moving several centimetres a year relative to WGS84. With USEREXPDATUM you can also input the velocity of the movement to account for drift over the years.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	USEREXPDATUM 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	semimajor	6300000.0 - 6400000.0 m		Datum semi-major axis (a) in metres	Double	8	H
3	flattening	290.0 - 305.0		Reciprocal Flattening, 1/f = a/(a-b)	Double	8	H+8
4	dx	± 2000.0 m		Datum offsets from local to WGS84. These are the translation values between the user datum and WGS84 (internal reference)	Double	8	H+16
5	dy	± 2000.0 m			Double	8	H+24
6	dz	± 2000.0 m			Double	8	H+32
7	rx	± 10.0 radians		Datum rotation angle about X, Y and Z. These values are the rotation from your local datum to WGS84. A positive sign is for counter clockwise rotation and a negative sign is for clockwise rotation	Double	8	H+40
8	ry	± 10.0 radians			Double	8	H+48
9	rz	± 10.0 radians			Double	8	H+56
10	scale	± 10.0 ppm		Scale value is the difference in ppm between the user datum and WGS84	Double	8	H+64
11	xvel	± 2000.0 m/yr		Velocity vector along X-axis	Double	8	H+72
12	yvel	± 2000.0 m/yr		Velocity vector along Y-axis	Double	8	H+80
13	zvel	± 2000.0 m/yr		Velocity vector along Z-axis	Double	8	H+88
14	xrvel	± 10.0 radians/yr		Change in the rotation about X over time	Double	8	H+96
15	yrvel	± 10.0 radians/yr		Change in the rotation about Y over time	Double	8	H+104
16	zrvel	± 10.0 radians/yr		Change in the rotation about Z over time	Double	8	H+112
17	scalev	± 10.0 ppm/yr		Change in scale from WGS84 over time	Double	8	H+120
18	refdate	0.0 year		Reference date of parameters Example: 2011.00 = Jan 1, 2011 2011.19 = Mar 11, 2011	Double	8	H+128

2.4.174 UTMZONE
Sets UTM parameters

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This command sets the UTM persistence, zone number or meridian. Refer to earth-info.nga.mil/GandG/coordsys/grids/referencesys.html for more information and a world map of UTM zone numbers.

> (i) 1. The latitude limits of the UTM System are 80°S to 84°N, so if your position is outside this range, the BESTUTM log outputs a northing, easting and height of 0.0, along with a zone letter of "*" and a zone number of 0, so that it is obvious that the data in the log is dummy data.
>
> 2. If the latitude band is X, then the Zone number should not be set to 32, 34 or 36. These zones were incorporated into other zone numbers and do not exist.

Abbreviated ASCII Syntax: **Message ID: 749**

UTMZONE command [parameter]

Factory Default:

utmzone auto 0

ASCII Example 1:

utmzone set 10

ASCII Example 2:

utmzone current

> (i) The UTM grid system is displayed on all National Topographic Series (NTS) of Canada maps and United States Geological Survey (USGS) maps. On USGS 7.5-minute quadrangle maps (1:24,000 scale), 15-minute quadrangle maps (1:50,000, 1:62,500, and standard-edition 1:63,360 scales) and Canadian 1:50,000 maps the UTM grid lines are drawn at intervals of 1,000 metres and are shown either with blue ticks at the edge of the map or by full blue grid lines. On USGS maps at 1:100,000 and 1:250,000 scale and Canadian 1:250,000 scale maps a full UTM grid is shown at intervals of 10,000 metres.

Table 71: UTM Zone Commands

Binary	ASCII	Description
0	AUTO	UTM zone default that automatically sets the central meridian and does not switch zones until it overlaps by the set persistence. This a spherical approximation to the earth unless you are at the equator (default = 0) (m)
1	CURRENT	Same as UTMZONE AUTO with infinite persistence of the current zone. The parameter field is not used
2	SET	Sets the central meridian based on the specified UTM zone. A zone includes its western boundary, but not its eastern boundary, Meridian. For example, zone 12 includes (108°W, 114°W) where 108° < longitude $\leq$ 114°
3	MERIDIAN	Sets the central meridian as specified in the parameter field. In BESTUTM, the zone number is output as 61 to indicate the manual setting (zones are set by pre-defined central meridians not user-set ones)

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	UTMZONE 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	command	See *Table 71, UTM Zone Commands* on *page 348*			Enum	4	H
3	parameter	See *Table 71, UTM Zone Commands* on *page 348*			Long	4	H+4

2.4.175 WIFIAPCONFIG
Configure the Wi-Fi AP

OEM Platform: ProPak6

This command configures the Wi-Fi Access Points (AP), which are used when the Wi-Fi radio is configured for use as an AP. Up to four APs can be configured on the ProPak6. Each AP supports up to 5 clients associated at one time. By default, ProPak6 AP ID 1 is enabled.

By default and after a factory reset (FRESET), the Wi-Fi AP is enabled. Refer to the ProPak6 User Manual (OM-20000148) for instructions on enabling Wi-Fi AP.

> (i) Also see the WIFICONFIG command (*page 356*) for more information.

When changing any settings using the WIFIAPCONFIG command, the Wi-Fi controller requires a power cycle before the settings will take effect. See the WIFICONFIG command (*page 356*) to change the state to OFF then back to disabled/enabled.

> (i) Use the DHCPCONFIG command to configure the optional parameters for the DHCP Server.

Abbreviated ASCII Syntax: **Message ID: 1665**

WIFIAPCONFIG [wifiapid] wifiapconfiguration value

ASCII Example:

wifiapconfig 1 ssid "myssid"
wifiapconfig 1 ipaddr "10.10.23.1"

> (i) This command must be entered in ASCII or Abbreviated ASCII.

Security

By default, minimal network/port security is set. It is the responsibility of the user to assess security requirements and configure the ProPak6 as necessary. Refer to the Security section of the ProPak6 User Manual (OM-20000148) for security settings.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	WIFIAPCONFIG header			Command header			0

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
2	wifiapid	1 2 3 4		The ID of the AP being configured. 1 = AP 1 2 = AP 2 3 = AP 3 4 = AP 4	Enum	4	H
3	wifiapconfiguration	*Table 72, WIFIAPCONFIG Parameters and Values* on *page 350*		Configuration parameter for the network.	Enum	4	H+4
4	value	*Table 72, WIFIAPCONFIG Parameters and Values* on *page 350*		The value assigned to the configuration parameter. The valid range for Value depends on which configuration parameter is being changed.	String [68]	variable[a]	H+8

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 72: WIFIAPCONFIG Parameters and Values

WiFiAP Configuration Parameter	Value	Description
SSID	STRING	The Service Set Identifier (network name) of the Wi-Fi AP.
IsHidden	TRUE or FALSE	When this parameter set to TRUE, the Wi-Fi AP does not broadcast the SSID.
Channel	802.11 Channel	The 802.11 Channel to use for this Wi-FI AP. The valid range of Channels depends on 802.11 regulatory domain. In North America the valid range is: 1-11
Authentication	OPEN WPA_PSK WPA2_PSK WPA/WPA2_PSK	The authentication type used for the Wi-Fi AP. The value for Authentication is dependent on the values for Encryption and Protocol. See *Table 73, Legal Combination of Authentication, Encryption, Protocol* on *page 351* for the valid combinations of Authentication, Encryption and Protocol.
Encryption	NONE TKIP AES_CCMP TKIP AES_CCMP	The encryption protocol used for the Wi-Fi AP. The value for Encryption is dependent on the values for Authentication and Protocol. See *Table 73, Legal Combination of Authentication, Encryption, Protocol* on *page 351* for the valid combinations of Authentication, Encryption and Protocol.
Passphrase	STRING	WPA/WPA2 passphrase (8 to 63 ASCII characters) or Hex key (32 bytes - 64 ASCII characters)
IPAddr	ddd.ddd.ddd.ddd (for example: 10.0.0.2)	The IP address of the Wi-Fi AP. This must be a static IP address.

WiFiAP Configuration Parameter	Value	Description
Netmask	ddd.ddd.ddd.ddd (for example: 255.255.255.0)	The netmask for Wi-Fi AP IP address.
Gateway	ddd.ddd.ddd.ddd (for example: 10.0.0.1)	The IP address of the gateway
Protocol	b g bg bgn	The 802.11 standard protocol used for the Wi-Fi AP. The value for Protocol is dependent on the values for Authentication and Encryption. See Table 73 for the valid combinations of Authentication, Encryption and Protocol.
Client_Timeout	1 to 3600 seconds	Time in seconds for the Wi-Fi AP to detect a lost client connection.

Table 73: Legal Combination of Authentication, Encryption, Protocol

Authentication	Encryption	Protocol	Comments
OPEN	NONE	b, g, bg, bgn	No security.
WPA_PSK	TKIP	b, g, bg	802.11n cannot be used with TKIP-only encryption Not recommended. Intended to support legacy clients only.
WPA2_PSK	AES_CCMP	b, bg, bgn	
WPA/WPA2_PSK	TKIP AES_CCMP	b, g, bg, bgn	Not recommended. Intended to support legacy clients only

Table 74: WIFIAPCONFIG Default Parameters for AP 1

Parameter	Value
SSID	ProPak6 <PSN>
IsHidden	FALSE
Channel	6
Authentication	WPA2_PSK
Encryption	AES_CCMP
Passphrase	<PSN>
IPAddr	192.168.1.1
Netmask	255.255.255.0
Protocol	bgn
Client_Timeout	300 seconds

For AP 2, AP 3 and AP 4, all of the `WIFIAPCONFIG` parameters are blank by default.

2.4.176 WIFICLICONFIG
Configures Wi-Fi client

OEM Platform: ProPak6

This command configures the Wi-Fi client. The ProPak6 can store the configurations for a maximum of 16 Access Points (AP).

> ⓘ To use the ProPak6 as an Wi-Fi client, use the WIFICONFIG command to set the Wi-Fi mode to client and Wi-Fi state to enabled.
>
>   wificonfig mode client
>
>   wificonfig state enabled
>
> See the `WIFICONFIG` command (*page 356*) for more information.

A minimal configuration includes setting the SSID, Authentication and Encryption (to connect to an open AP using DHCP). Contact your network administrator for the settings required on your local network architecture.

Abbreviated ASCII Syntax: **Message ID: 1614**

WIFICLICONFIG wifinetworkid wifinetworkconfiguration value

Factory Defaults:

SSID =	blank
Passphrase =	blank
Authentication =	OPEN
Encryption =	NONE
DHCP =	TRUE

ASCII Example:

        wificliconfig 1 ssid novatel

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	WIFICLICONFIG 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	wifinetworkid	ID of the Wi-Fi network Valid range: 1 to 16	Enum	4	H
3	wifinetwork configuration	Configuration parameter for the network See *Table 75, WIFICLICONFIG Parameters* on *page 353*	Enum	4	H+4
4	value	Value for the wifinetworkconfiguration parameter See *Table 75, WIFICLICONFIG Parameters* on *page 353*	String [68]	variable[a]	H+8

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 75: WIFICLICONFIG Parameters

WiFiNetwork Configuration	Value	Description
SSID	STRING	Service Set Identifier of the Wi-Fi AP being connected to This value must be entered in ASCII characters only (no binary).
BSSID	aa::bb::cc::ee::ff	Basic Service Set ID. The client will connect only to this BSSID.
Enabled	TRUE, FALSE	When set to TRUE, the Wi-Fi client network is included as a potential connection when the client interface is enabled and the system is searching for Access Points it knows. When set to FALSE, the Wi-Fi client network is kept for reference but is not automatically used.
Priority	ONT	Instructs the client to always attempt connecting to this network, regardless of availability of other networks, signal strength, etc
Authentication	OPEN WPA_PSK WPA2_PSK	The authentication method for the AP being connected to.
Encryption	None TKIP AES_CCMP	The encryption method for the AP being connected to.
Passphrase	String, 64	WPA, WPA2 passphrase (8-63 ascii chars), or key (64 hex digits)
DHCP	TRUE, FALSE	When set to TRUE, DHCP is used to acquire the network configuration from the AP. When set to FALSE the static configuration values (shown below) are used to connect to the AP. Note that unless this value is specifically set to FALSE, DHCP is enabled regardless of any network parameters that may be set (IP, Netmask, Gateway, DNS)
IPAddr	ddd.ddd.ddd.ddd (for example: 10.0.0.2)	The IP address to be used by the receiver. (If not using DHCP.)
Netmask	ddd.ddd.ddd.ddd (for example: 255.255.255.0)	The netmask to be used by the receiver. (If not using DHCP.)
Gateway	ddd.ddd.ddd.ddd (for example: 10.0.0.1)	The IP address of the default gateway. (If not using DHCP.)
DNS1	ddd.ddd.ddd.ddd (for example: 10.0.0.3)	The IP address of the primary DNS server. (If not using DHCP.)

2.4.177 WIFICLICONTROL
Controls Wi-Fi client

OEM Platform: ProPak6

Use the `WIFILCLICONTROL` command to scan for Access Points (AP) within range of the Propak6 or to select one of the configurations already entered (using the WIFICLICONFIG command, see *page 352*) to gain access to a particular AP.

Any changes resulting from `WIFICLICONTROL` are transient, i.e., they only affect the running instance of the Wi-Fi stack and do not persist across restarts of the Wi-Fi stack.

> ▼ The `WIFILCLICONTROL` command cannot be saved with `SAVECONFIG`.

Abbreviated ASCII Syntax: **Message ID: 1615**

WIFICLICONTROL wificlientcontrol value

ASCII Example:

wificlicontrol scan

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	WIFICLICONTROL header	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively			0
2	wificlientcontrol	Control parameter (refer to *Table 76, WIFICLICONTROL Parameters* on *page 355*)	Enum	4	H
3	value	Used by the control parameter	String [32]	variable[a]	H+4

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

> ⓘ If you experience problems connecting to a new Wi-Fi Access Point when previously connected to another Access Point, the Wi-Fi module may need to be restarted. Issue the following commands to restart the Wi-Fi module:
>
> wificonfig state off
>
> log wificlistatusa onchanged   (wait for wificlistatus to show off)
>
> wificonfig state enabled

Table 76: WIFICLICONTROL Parameters

WifiClientControl	Description
SCAN	Triggers a scan to find any APs within range of the ProPak6. Results are output into the WIFICLISCANRESULTS log (see *page 750*) and can be used to configure Wi-Fi networks using the `WIFICLICONFIG` command
APPLYCONFIG	Applies configuration changes made with `WIFICLICONFIG` and enables automatic connectivity. This may result in the network disconnecting and reconnecting.
DISCONNECT	Disconnects from the current AP, if any. Automatic connectivity is disabled.
CONNECT	Connects to one of the networks defined using the WIFICLICONFIG command. This parameter requires a value. The valid values are: 1 - 16

2.4.178 WIFICONFIG
Configure the Wi-Fi radio power and operating mode

OEM Platform: ProPak6

Use this command to configure the power and operating mode of the Wi-Fi radio. When the Wi-Fi radio is enabled, it can run in one of two states:

- Client mode
- Access Point (AP) mode

Client mode and AP mode are mutually exclusive. The AP settings are configured using the WIFIAPCONFIG command (see *page 349*). The Client settings are configured using the WIFICLICONFIG command (see *page 352*).

When the WIFICONFIG command is used to change the operational MODE to AP or Client, the Wi-Fi controller powers on and the radio is automatically set to the enabled state.

> (i) This command can be saved with the SAVECONFIG command.

Abbreviated ASCII Syntax: **Message ID: 1617**

WIFICONFIG wificonfiguration parameter 1 [parameter 2]

Factory Default:

> wificonfig mode ap
> wificonfig state enabled

> (i) The factory default sets the Wi-Fi radio to AP mode enabled. See *Table 74, WIFIAPCONFIG Default Parameters for AP 1* on *page 351* for the default AP mode settings.

ASCII Examples (AP):

wificonfig mode ap	Configures the receiver as an AP with the default profile (1)
wificonfig mode ap 2	Configures the receiver as AP with AP profile 2
wificonfig state enabled	Enables Wi-Fi as an AP (since the mode is set to AP)
wificonfig state disabled	Disables Wi-Fi
wificonfig state off	Completely powers off 802.11 chip

ASCII Examples (client):

wificonfig mode client	Configures the receiver as a client.
wificonfig state enabled	Enables Wi-Fi as a client (since the mode is set to client)
wificonfig state disabled	Disables Wi-Fi
wificonfig state off	Completely powers off 802.11 chip

> ⚠ Changing the Wi-Fi mode (from AP to Client or Client to AP) resets the Wi-Fi and Bluetooth radio. Any active Bluetooth connections are terminated.

Field	Field Type	ASCII Value	Binary Value	Description	Format	Binary Bytes	Binary Offset
1	WIFICONFIG 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	wificonfiguration	MODE	1	Set the operating mode of the Wi-Fi radio	Enum	4	H
		STATE	4	Set the state of the Wi-FI radio			
3	parameter 1	The valid values for parameter 1 depend on whether MODE or STATE was selected. See *Table 77, WIFICONFIG Parameters* on *page 357*			String [32]	variable[a]	H + 4
4	parameter 2	The valid values for parameter 2 depend on whether MODE or STATE was selected. See *Table 77, WIFICONFIG Parameters* on *page 357*			String [32]	variable[a]	variable

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 77: WIFICONFIG Parameters

wificonfiguration	Parameter 1 String	Parameter 2 String	Description
MODE	CLIENT	N/A	Sets the Wi-Fi radio to Client mode (802.11 STA)
MODE	AP	1 2 3 4	Sets the Wi-Fi radio to AP mode (802.11 Infrastructure Access Point) and selects the AP Profile. The AP Profile can be 1, 2, 3 or 4. The default value for the AP Profile is 1.
STATE	ENABLED (default)	N/A	Enables the MODE, Wi-Fi radio is active Powers on the 802.11 hardware, if it was powered off using STATE OFF
STATE	DISABLED	N/A	Disables the MODE, Wi-Fi radio is inactive
STATE	OFF[a]	N/A	Disables the MODE, Wi-Fi radio is inactive, 802.11 hardware powered off The 802.11 firmware is reset the next time the Wi-Fi radio is powered on

a. Typically, DISABLED is used rather than OFF.
   OFF is required only if there is a specific need to power off the 802.11 hardware.

Chapter 3 Data Logs

3.1   Log Types

See the LOG command on *page 202*, for details about requesting logs.

The receiver is capable of generating three type of logs: synchronous, asynchronous and polled. The data for synchronous logs is generated on a regular schedule. In order to output the most current data as soon as it is available, asynchronous data is generated at irregular intervals. The data in polled logs is generated on demand. The following table outlines the log types and the valid triggers to use:

Table 78: Log Type Triggers

Type	Recommended Trigger	Illegal Trigger
Synch	ONTIME	ONNEW, ONCHANGED
Asynch	ONCHANGED or ONCE	-
Polled	ONCE or ONTIME [a]	ONNEW, ONCHANGED

a. Polled log types do not allow fractional offsets and cannot do ontime rates faster than 1 Hz.

See *Section 1.5, Message Time Stamps* on *page 36* for information about how the message time stamp is set for each type of log.

1. The OEM6 family of receivers can handle 64 logs at a time. If an attempt is made to log more than 64 logs at a time, the receiver responds with an Insufficient Resources error.
2. The following logs do not support the ONNEXT trigger: GPSEPHEM, RAWEPHEM, RAWGPSSUBFRAME, RAWSBASFRAME, RXSTATUSEVENT and SBAS9.
3. Asynchronous logs, such as MATCHEDPOS, should only be logged ONCHANGED. Otherwise, the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may result in inaccurate time tags.
4. Use the ONNEW trigger with the MARKTIME or MARKPOS logs.
5. Before the output of fields for ASCII and binary logs, there is an ASCII or binary header respectively. See also *Table 2, ASCII Message Header Structure* on *page 24* and *Table 3, Binary Message Header Structure* on *page 26*. There is no header information before Abbreviated ASCII output, see *page 25*.

3.1.1   Log Type Examples

For polled logs, the receiver only supports an offset that is:

- smaller than the logging period
- an integer

The following are valid examples for a polled log:

> log comconfig ontime 2 1
>
> log portstats ontime 4 2
>
> log version once

For polled logs, the following examples are invalid:

> log comconfig ontime 1 2[offset is larger than the logging period]
>
> log comconfig ontime 4 1.5[offset is not an integer]

For synchronous and asynchronous logs, the receiver supports any offset that is:

- smaller than the logging period
- a multiple of the minimum logging period

For example, if the receiver supports 20 Hz logging, the minimum logging period is 1/20 Hz or 0.05 s. The following are valid examples for a synchronous or asynchronous log, on a receiver that can log at rates up to 20 Hz:

> log bestpos ontime 1[1 hz]
>
> log bestpos ontime 1 0.1
>
> log bestpos ontime 1 0.90
>
> log avepos ontime 1 0.95
>
> log avepos ontime 2[0.5 hz]
>
> log avepos ontime 2 1.35
>
> log avepos ontime 2 1.75

For synchronous and asynchronous logs, the following examples are invalid:

> log bestpos ontime 1 0.08[offset is not a multiple of the minimum logging period]
>
> log bestpos ontime 1 1.05[offset is larger than the logging period]

3.2 Log Reference

Table 79: OEM6 Logs By Function

LOGS	DESCRIPTIONS	TYPE
GENERAL RECEIVER CONTROL AND STATUS		
APPLICATIONSTATUS	Provides application status information	Asynch
AUTHCODES	Contains all authorization codes (auth codes) entered into the system since the last complete firmware reload	Polled
BLUETOOTHSTATUS	Bluetooth radio module status	Asynch
CELLULARINFO	Cellular modem and network information	Asynch
CELLULARSTATUS	Cellular modem and network status information	Asynch
DIRENT	Onboard memory file list	Polled
ETHSTATUS	Current Ethernet status	Asynch
HWMONITOR	Monitor hardware levels	Polled
IPSTATS	IP statistics	Polled
IPSTATUS	Current network configuration status	Asynch
LOGFILESTATUS	Current state of file and recording	Asynch
LOGLIST	List of system logs	Polled
PASSAUX, PASSCOM1, PASSCOM2, PASSCOM3, PASSCOM4, PASSCOM5, PASSCOM6, PASSCOM7, PASSCOM8, PASSCOM9, PASSCOM10, PASSETH1, PASSICOM1, PASSICOM2, PASSICOM3, PASSNCOM1, PASSNCOM2, PASSNCOM3, PASSUSB1, PASSUSB2, PASSUSB3, PASSXCOM1, PASSXCOM2, PASSXCOM3	Pass-through log that redirects data from one port to another port	Asynch
PASSTHROUGH	Outputs pass-through data from all receiver ports	Asynch
PORTSTATS	Displays port statistics	Polled
PROFILEINFO	Outputs a list of profiles	Polled
RTCAOBS3	Proprietary message that carries dual-frequency GPS and GLO measurements and is used in ALIGN. Also carries SBAS measurements if the Master receiver is single-frequency (L1-only) receiver to enable SBAS-ALIGN at the L1-only ALIGN Rover	Synch
RXCONFIG	Receiver configuration status	Polled
RXSTATUS	Self-test status	Asynch
RXSTATUSEVENT	Status event indicator	Asynch

LOGS	DESCRIPTIONS	TYPE
SOURCETABLE	Outputs the NTRIP source table entries from the NTRIPCASTER set by the NTRIPSOURCETABLE command	Synch
VALIDMODELS	Model and expiry date information for receiver	Asynch
VERSION	Receiver hardware and software version numbers	Polled
WIFIAPSTATUS	Wi-Fi Access Point Status	Asynch
WIFICLISCANRESULTS	Wi-Fi AP scan results	Asynch
WIFICLISTATUS	Wi-Fi client connection status	Asynch
POSITION, PARAMETERS AND SOLUTION FILTERING CONTROL		
ALIGNBSLNXYZ	Outputs the RTK quality XYZ baselines from ALIGN.	Asynch
ALIGNBSLNENU	Outputs the RTK quality ENU baselines from ALIGN. The XYZ baselines (output in ALIGNBSLNXYZ log) are rotated relative to master position (output in MASTERPOS) to compute ENU baselines	Asynch
ALIGNDOP	Outputs the DOP computed using the satellites used in solution	Asynch
AVEPOS	Position averaging log	Asynch
BESTPOS [a]	Best position data	Synch
BESTUTM	Best available UTM data	Synch
BESTXYZ	Cartesian coordinates position data	Synch
BSLNXYZ	RTK XYZ baseline	Synch
GALIONO	Decoded Galileo ionospheric corrections	Asynch
GPGGA	NMEA, fix and position data	Synch
GPGGALONG	GPS Fix Data, Extra Precision and undulation	Synch
GPGGARTK	NMEA, global position system fix data	Synch
GPGLL	NMEA, position data	Synch
GPGRS	NMEA, range residuals	Synch
GPGSA	NMEA, DOP information	Synch
GPGST	NMEA, measurement noise statistics	Synch
GPHDT	NMEA, heading from True North	Asynch
HEADING	Heading information with the ALIGN feature	Asynch
HEADING2	Outputs same information as HEADING log with an additional Rover ID field	Asynch
HEADINGRATE	Provides rate of change for the heading parameters	Asynch
HEADINGSATS	Outputs the satellite information from ALIGN filter	Asynch
IONUTC	Ionospheric and UTC model information	Asynch
MARKPOS, MARK2POS	Position at time of mark input event	Asynch

LOGS	DESCRIPTIONS	TYPE
MARKTIME, MARK2TIME, MARK3TIME, MARK4TIME	Time of mark input event	Asynch
MASTERPOS	Displays the master position with the ALIGN feature	Asynch
MATCHEDPOS [a]	Computed position	Asynch
MATCHEDXYZ	Cartesian coordinates computed position data	Asynch
OMNIHPPOS	OmniSTAR HP/XP/G2 position data	Synch
PDPPOS	PDP filter position	Synch
PDPVEL	PDP filter velocity	Synch
PDPXYZ	PDP filter Cartesian position and velocity	Synch
PPPPOS	PPP filter position	Synch
PSRDOP	DOP of SVs currently tracking	Asynch
RAIMSTATUS	RAIM status	Synch
ROVERPOS	Displays the rover position with the ALIGN feature	Asynch
RTKDOP	Values from the RTK fast filter	Synch
RTKDOP2	Values from the RTK fast filter	Synch
RTKPOS [a]	RTK low latency position	Synch
RTKVEL[b]	RTK velocity	Synch
RTKXYZ	RTK Cartesian coordinate position	Synch

a. The RTK system in the receiver provides two kinds of position solutions. The Matched RTK position is computed with buffered observations, so there is no error due to the extrapolation of base station measurements. This provides the highest accuracy solution possible at the expense of some latency which is affected primarily by the speed of the differential data link. The MATCHEDPOS log contains the matched RTK solution and can be generated for each processed set of base station observations.

The Low-Latency RTK position is computed from the latest local observations and extrapolated base station observations. This supplies a valid RTK position with the lowest latency possible at the expense of some accuracy. The degradation in accuracy is reflected in the standard deviation and is summarized in our *An Introduction to GNSS* book, available through our website at www.novatel.com through *Support | Knowledge and Learning > Learn About GNSS > GNSS Book*. The amount of time that the base station observations are extrapolated is in the "differential age" field of the position log. The Low-Latency RTK system extrapolates for 60 seconds. The RTKPOS log contains the Low-Latency RTK position when valid, and an "invalid" status when a Low-Latency RTK solution could not be computed. The BESTPOS log contains either the low-latency RTK, PPP, OmniSTAR HP/XP or pseudorange-based position, whichever has the smallest standard deviation.

LOG	DESCRIPTION	TYPE
WAYPOINT NAVIGATION		
BESTPOS	Best position data	Synch

b. The RTK velocity is computed from successive Low-Latency RTK position solutions. The RTKVEL log contains the RTK velocity, when valid and outputs an 'invalid' status if a Low-Latency RTK velocity solution cannot be computed. The BESTVEL log contains the Low-Latency RTK velocity when the BESTPOS log contains the Low-Latency RTK position.

In a BESTVEL, PSRVEL or RTKVEL log, the actual speed and direction of the receiver antenna over ground is provided. The receiver does not determine the direction a vessel, craft or vehicle is pointed (heading) but rather the direction of motion of the GNSS antenna relative to ground.

LOG	DESCRIPTION	TYPE
WAYPOINT NAVIGATION		
BESTVEL[a]	Velocity data	Synch
GPHDT	NMEA, heading from True North	Asynch
GPRMB	NMEA, waypoint status	Synch
GPRMC	NMEA, navigation information	Synch
GPVTG	NMEA, track made good and speed	Synch
NAVIGATE	Navigation waypoint status	Synch

a. In a BESTVEL, PSRVEL or RTKVEL log, the actual speed and direction of the receiver antenna over ground is provided. The receiver does not determine the direction a vessel, craft or vehicle is pointed (heading) but rather the direction of motion of the GNSS antenna relative to ground.

LOG	DESCRIPTION	TYPE
CLOCK INFORMATION, STATUS AND TIME		
CLOCKMODEL	Range bias information	Synch
CLOCKSTEERING	Clock steering status	Asynch
GALCLOCK	Galileo time information	Asynch
GLOCLOCK	GLONASS clock information	Asynch
GPZDA	NMEA, UTC time and data	Synch
MARK1COUNT	Count for the Mark1 input	Asynch
MARK2COUNT	Count for the Mark2 input	Asynch
MARK3COUNT	Count for the Mark3 input	Asynch
MARK4COUNT	Count for the Mark4 input	Asynch
PSRTIME	Time offsets from the pseudorange filter	Synch
TIME	Receiver time information	Synch
TIMESYNC	Synchronize time between receivers	Synch
POST PROCESSING DATA		
GPSEPHEM	Decoded GPS ephemeris information	Asynch
IONUTC	Ionospheric and UTC model information	Asynch

LOG	DESCRIPTION	TYPE
RANGE	Satellite range information	Synch
RANGECMP	Compressed version of the RANGE log	Synch
RANGECMP2	RANGE data compressed to handle more channels and types	Synch
RANGEGPSL1	L1 version of the RANGE log	Synch
RAWEPHEM	Raw ephemeris	Asynch
TIME	Receiver clock offset information	Synch
SATELLITE TRACKING AND CHANNEL CONTROL		
ALMANAC	Current decoded almanac data	Asynch
BDSALMANAC	Decoded almanac parameters as received from the satellite, with the parity information removed and appropriate scaling applied	Asynch
BDSCLOCK	Time parameters transmitted by the BeiDou satellites	Asynch
BDSEPHEMERIS	A single set of BDS ephemeris parameters	Asynch
BDSIONO	Contains the Klobuchar ionosphere model parameters transmitted by the BeiDou satellites	Asynch
BDSRAWNAVSUBFRAME	Log contains single set of BDS ephemeris parameters	Asynch
BESTSATS	Satellites used in BESTPOS	Synch
CHANCONFIGLIST	Channel configuration list	Polled
GALALMANAC	Decoded Galileo almanac parameters from Galileo navigation messages	Asynch
GALEPHEMERIS	Galileo ephemeris information is available through the GALEPHEMERIS log	Asynch
GALFNAVRAWPAGE	Contains the raw Galileo F/Nav page data	Asynch
GALINAVRAWWORD	Contains the raw Galileo I/Nav word data	Asynch
GLMLA	NMEA GLONASS almanac data	Asynch
GLOALMANAC	GLONASS almanac data	Asynch
GLOEPHEMERIS	GLONASS ephemeris data	Asynch
GLORAWALM	Raw GLONASS almanac data	Asynch
GLORAWEPHEM	Raw GLONASS ephemeris data	Asynch
GLORAWFRAME	Raw GLONASS frame data	Asynch
GLORAWSTRING	Raw GLONASS string data	Asynch
GPALM	NMEA, almanac data	Asynch
GPGSA	NMEA, SV DOP information	Synch
GPGSV	NMEA, satellite-in-view information	Synch

LOG	DESCRIPTION	TYPE
GPSEPHEM	Decoded GPS ephemeris information	Asynch
LBANDBEAMTABLE	List of L-Band Beams	Asynch
LBANDTRACKSTAT	L-Band Tracking Status	Synch
MATCHEDSATS	Lists the used and unused satellites for the corresponding MATCHEDPOS solution	Asynch
OMNIHPSATS	Satellites used in the OMNIHPPOS solution	Synch
OMNIVIS	OmniSTAR satellite visibility list	Synch
PDPSATS	Satellites used in PDPPOS solution	Synch
PSRDOP	DOP of SVs currently tracking	Asynch
PSRDOP2	Pseudorange Least Squares DOP	Asynch
PSRSATS	Satellites used in PSRPOS solution	Synch
QZSSALMANAC	Contains the decoded almanac parameters as received from the satellite with the parity information removed and appropriate scaling applied	Asynch
QZSSEPHEMERIS	Single set of QZSS ephemeris parameters	Asynch
QZSSIONUTC	Ionospheric Model parameters (ION) and the Universal Time Coordinated parameters (UTC) for QZSS are provided	Asynch
QZSSRAWALMANAC	Contains the undecoded almanac subframes as received from the QZSS satellite	Asynch
QZSSRAWEPHEM	Contains the raw binary information for subframes one, two and three from the satellite with the parity information removed	Asynch
QZSSRAWSUBFRAME	A raw QZSS subframe is 300 bits in total, includes the parity bits which are interspersed with the raw data ten times, in six bit chunks, for a total of 60 parity bits	Asynch
RANGE	Satellite range information	Synch
RANGEGPSL1	L1 version of the RANGE log	Synch
RAWALM	Raw almanac	Asynch
RAWCNAVFRAME	Raw L2C frame data	Asynch
RAWEPHEM	Raw ephemeris	Asynch
RAWGPSSUBFRAME	Raw subframe data	Asynch
RAWGPSWORD	Raw navigation word	Asynch
RAWSBASFRAME	Raw SBAS frame data	Asynch
RTKSATS	Satellites used in RTKPOS solution	Synch
SATVIS	Satellite visibility	Synch
SATVIS2	Satellite visibility	Asynch

LOG	DESCRIPTION	TYPE
SATXYZ2	Combined with a RANGE log, contains the decoded satellite information necessary to compute the solution	Synch
SBAS0	Removes PRN from the solution	Asynch
SBAS1	PRN mask assignments	Asynch
SBAS2	Fast correction slots 0-12	Asynch
SBAS3	Fast correction slots 13-25	Asynch
SBAS4	Fast correction slots 26-38	Asynch
SBAS5	Fast correction slots 39-50	Asynch
SBAS6	Integrity message	Asynch
SBAS7	Fast correction degradation	Asynch
SBAS9	GEO navigation message	Asynch
SBAS10	Degradation factor	Asynch
SBAS12	SBAS network time and UTC	Asynch
SBAS17	GEO almanac message	Asynch
SBAS18	IGP mask	Asynch
SBAS24	Mixed fast/slow corrections	Asynch
SBAS25	Long-term slow satellite corrections	Asynch
SBAS26	Ionospheric delay corrections	Asynch
SBAS27	SBAS service message	Asynch
SBAS32	Fast correction slots 0-10	Asynch
SBAS33	Fast correction slots 11-21	Asynch
SBAS34	Fast correction slots 22-32	Asynch
SBAS35	Fast correction slots 39-50	Asynch
SBAS45	Slow corrections	Asynch
SBASALMANAC	A collection of all current SBAS almanacs decoded by the receiver	Asynch
SBASCORR	SBAS range corrections used	Synch
SOFTLOADSTATUS	Describes the status of the SoftLoad process	Asynch
TERRASTARINFO	TerraStar Subscription Information	Asynch
TERRASTARSTATUS	TerraStar Decoder and Subscription Status	Asynch
TRACKSTAT	Satellite tracking status	Synch
VERIPOSINFO	Veripos Subscription Information	Asynch
VERIPOSSTATUS	Veripos Decoder and Subscription Status	Asynch

LOG	DESCRIPTION	TYPE
	DIFFERENTIAL BASE STATION	
ALMANAC	Current almanac information	Asynch
BESTPOS	Best position data	Synch
BESTVEL	Velocity data	Synch
BSLNXYZ	RTK XYZ baseline	Asynch
GPGGA	NMEA, position fix data	Synch
GPGGARTK	NMEA, global position system fix data	Synch
LBANDINFO	L-Band configuration information	Asynch
LBANDSTAT	L-Band status information	Synch
MATCHEDPOS	Computed position – Time Matched	Asynch
OMNIHPPOS	OmniSTAR HP/XP/G2 position data	Synch
PSRPOS	Pseudorange position	Synch
PSRVEL	Pseudorange velocity	Synch
RANGE	Satellite range information	Synch
RANGECMP	Compressed version of the RANGE log	Synch
RAWLBANDFRAME	Raw L-Band frame data	Asynch
RAWLBANDPACKET	Raw L-Band data packet	Asynch
REFSTATION	Base station position and health	Asynch
REFSTATIONINFO	Reference station position and health	Asynch
RTCA1	Type 1 Differential GPS corrections	Synch
RTCAEPHEM	Type 7 Ephemeris and time information	Synch
RTCAOBS	Type 7 Base Station observations	Synch
RTCAOBS2	Type 7 Base Station observations 2	Synch
RTCAREF	Type 7 Base Station parameters	Synch
RTCAREFEXT	Type 7 Extended Base Station parameters	Synch
RTKPOS	RTK low latency position	Synch

Table 80: OEM6 Logs in Alphabetical Order

DATA TYPE	MESSAGE ID	DESCRIPTION
ALIGNBSLNENU	1315	Outputs the RTK quality ENU baselines from ALIGN
ALIGNBSLNXYZ	1314	Outputs the RTK quality XYZ baselines from ALIGN
ALIGNDOP	1332	Outputs the DOP computed using the satellites used in solution
ALMANAC	73	Current almanac information

DATA TYPE	MESSAGE ID	DESCRIPTION
APPLICATIONSTATUS	520	Provides application status information
AUTHCODES	1348	Contains all authorization codes (auth codes) entered into the system since the last complete firmware reload
AVEPOS	172	Position averaging
BDSALMANAC	1584	Decoded almanac parameters as received from the satellite, with the parity information removed and appropriate scaling applied
BDSCLOCK	1607	Time parameters transmitted by the BeiDou satellites
BDSEPHEMERIS	1696	A single set of BDS ephemeris parameters
BDSIONO	1590	Contains the Klobuchar ionosphere model parameters transmitted by the BeiDou satellites
BDSRAWNAVSUBFRAME	1695	Log contains single set of BDS ephemeris parameters
BESTPOS	42	Best position data
BESTSATS	1194	Satellites used in BESTPOS
BESTUTM	726	Best available UTM data
BESTVEL	99	Velocity data
BESTXYZ	241	Cartesian coordinate position data
BLUETOOTHSTATUS	1608	Bluetooth radio module status
BSLNXYZ	686	RTK XYZ baseline
CELLULARINFO	1686	Cellular modem and network information
CELLULARSTATUS	1685	Cellular modem and network status information
CHANCONFIGLIST	1148	Channel configuration list
CLOCKMODEL	16	Current clock model matrices
CLOCKSTEERING	26	Clock steering status
DIRENT	159	Onboard memory file list
ETHSTATUS	1288	Current Ethernet status
GALALMANAC	1120	Decoded Galileo almanac parameters from Galileo navigation messages
GALCLOCK	1121	Galileo time information
GALEPHEMERIS	1122	Galileo ephemeris information is available through the GALEPHEMERIS log
GALFNAVRAWPAGE	1413	Contains the raw Galileo F/Nav page data
GALINAVRAWWORD	1414	Contains the raw Galileo I/Nav word data
GALIONO	1127	Decoded Galileo ionospheric corrections
GLOALMANAC	718	GLONASS almanac data

DATA TYPE	MESSAGE ID	DESCRIPTION
GLOCLOCK	719	GLONASS clock information
GLOEPHEMERIS	723	GLONASS ephemeris data
GLORAWALM	720	Raw GLONASS almanac data
GLORAWEPHEM	792	Raw GLONASS ephemeris data
GLORAWFRAME	721	Raw GLONASS frame data
GLORAWSTRING	722	Raw GLONASS string data
GPSEPHEM	7	GPS ephemeris data
HEADING	971	Heading information with the ALIGN feature
HEADING2	1335	Outputs same information as HEADING log with an additional Rover ID field
HEADINGRATE	1698	Provides rate of change for the heading parameters
HEADINGSATS	1316	Outputs the satellite information from ALIGN filter
HWMONITOR	963	Monitor hardware levels
IONUTC	8	Ionospheric and UTC model information
IPSTATS	1669	IP statistics
IPSTATUS	1289	Current network configuration status
LBANDBEAMTABLE	1718	List of L-Band Beams
LBANDINFO	730	L-Band configuration information
LBANDSTAT	731	L-Band status information
LBANDTRACKSTAT	1201	L-Band Tracking Status
LOGFILESTATUS	1146	Current state of file and recording
LOGLIST	5	A list of system logs
MARK1COUNT	1093	Count for the Mark1 input
MARK2COUNT	1094	Count for the Mark2 input
MARK3COUNT	1095	Count for the Mark3 input
MARK4COUNT	1096	Count for the Mark4 input
MARKPOS	181	Position at time of mark1 input event
MARK2POS	615	Position at time of mark2 input event
MARKTIME	231	Time of mark1 input event
MARK2TIME	616	Time of mark2 input event
MARK3TIME	1075	Time of mark3 input event
MARK4TIME	1076	Time of mark4 input event
MASTERPOS	1051	Displays master position with the ALIGN feature

DATA TYPE	MESSAGE ID	DESCRIPTION
MATCHEDPOS	96	RTK Computed Position – Time Matched
MATCHEDSATS	1176	Lists the used and unused satellites for the corresponding MATCHEDPOS solution
MATCHEDXYZ	242	RTK Time Matched cartesian coordinate position data
NAVIGATE	161	Navigation waypoint status
NOVATELXOBS	1618	NovAtel proprietary RTK correction
NOVATELXREF	1620	NovAtel proprietary reference station message for use in ALIGN
OMNIHPPOS	495	OmniSTAR HP/XP/G2 position data
OMNIHPSATS	1197	Satellites used in the OMNIHPPOS solution
OMNIVIS	860	OmniSTAR satellite visibility list
PASSAUX, PASSCOM1, PASSCOM2, PASSCOM3, PASSCOM4, PASSCOM5, PASSCOM6, PASSCOM7, PASSCOM8, PASSCOM9, PASSCOM10, PASSETH1, PASSICOM1, PASSICOM2, PASSICOM3, PASSNCOM1, PASSNCOM2, PASSNCOM3, PASSUSB1, PASSUSB2, PASSUSB3, PASSXCOM1, PASSXCOM2, PASSXCOM3	690, 233, 234, 235, 1384, 1576, 1577, 1701, 1702, 1703, 1704, 1209, 1250, 1251, 1252, 1253, 1254, 1255, 607, 608, 609, 405, 406, 795	Pass-through logs
PASSTHROUGH	1342	Outputs pass-through data from all receiver ports
PDPPOS	469	PDP filter position
PDPSATS	1234	Satellites used in PDPPOS solution
PDPVEL	470	PDP filter velocity
PDPXYZ	471	PDP filter Cartesian position and velocity
PORTSTATS	72	COM or USB port statistics
PPPPOS	1538	PPP filter position
PPPSATS	1541	Satellites used in the PPPPOS solution
PROFILEINFO	1412	Outputs a list of profiles
PSRDOP	174	DOP of SVs currently tracking
PSRDOP2	1163	Pseudorange Least Squares DOP
PSRPOS	47	Pseudorange position information

DATA TYPE	MESSAGE ID	DESCRIPTION
PSRSATS	1162	Satellites used in PSRPOS solution
PSRTIME	881	Time offsets from the pseudorange filter
PSRVEL	100	Pseudorange velocity information
PSRXYZ	243	Pseudorange Cartesian coordinate position information
QZSSALMANAC	1346	Contains the decoded almanac parameters as received from the satellite with the parity information removed and appropriate scaling applied
QZSSEPHEMERIS	1336	Single set of QZSS ephemeris parameters
QZSSIONUTC	1347	Ionospheric Model parameters (ION) and the Universal Time Coordinated parameters (UTC) for QZSS are provided
QZSSRAWALMANAC	1345	Contains the undecoded almanac subframes as received from the QZSS satellite
QZSSRAWEPHEM	1330	Contains the raw binary information for subframes one, two and three from the satellite with the parity information removed
QZSSRAWSUBFRAME	1330	A raw QZSS subframe is 300 bits in total, includes the parity bits which are interspersed with the raw data ten times, in six bit chunks, for a total of 60 parity bits
RAIMSTATUS	1286	RAIM status
RANGE	43	Satellite range information
RANGECMP	140	Compressed version of the RANGE log
RANGECMP2	1273	RANGE data compressed to handle more channels and types
RANGEGPSL1	631	L1 version of the RANGE log
RAWALM	74	Raw almanac
RAWCNAVFRAME	1066	Raw L2C frame data
RAWEPHEM	41	Raw ephemeris
RAWGPSSUBFRAME	25	Raw subframe data
RAWGPSWORD	407	Raw navigation word
RAWLBANDFRAME	732	Raw L-Band frame data
RAWLBANDPACKET	733	Raw L-Band data packet
RAWSBASFRAME	973	Raw SBAS frame data
REFSTATION	175	Base station position and health
REFSTATIONINFO	1325	Reference station position and health information
ROVERPOS	1052	Displays over position with the ALIGN feature
RTKDOP	952	Values from the RTK fast filter

DATA TYPE	MESSAGE ID	DESCRIPTION
RTKDOP2	1172	Values from the RTK fast filter
RTKPOS	141	RTK low latency position data
RTKSATS	1174	Satellites used in RTKPOS solution
RTKVEL	216	RTK velocity
RTKXYZ	244	RTK Cartesian coordinate position data
RXCONFIG	128	Receiver configuration status
RXSTATUS	93	Self-test status
RXSTATUSEVENT	94	Status event indicator
SATVIS	48	Satellite visibility
SATVIS2	1043	Satellite visibility
SATXYZ2	1451	Combined with a RANGE log, this data set contains the decoded satellite information necessary to compute the solution
SBAS0	976	Remove PRN from the solution
SBAS1	977	PRN mask assignments
SBAS2	982	Fast correction slots 0-12
SBAS3	987	Fast correction slots 13-25
SBAS4	992	Fast correction slots 26-38
SBAS5	994	Fast correction slots 39-50
SBAS6	995	Integrity message
SBAS7	996	Fast correction degradation
SBAS9	997	GEO navigation message
SBAS10	978	Degradation factor
SBAS12	979	SBAS network time and UTC
SBAS17	980	GEO almanac message
SBAS18	981	IGP mask
SBAS24	983	Mixed fast/slow corrections
SBAS25	984	Long term slow satellite corrections
SBAS26	985	Ionospheric delay corrections
SBAS27	986	SBAS service message
SBAS32	988	Fast correction slots 0-10
SBAS33	989	Fast correction slots 11-21
SBAS34	990	Fast correction slots 22-32
SBAS35	991	Fast correction slots 39-50

DATA TYPE	MESSAGE ID	DESCRIPTION
SBAS45	993	Slow corrections
SBASALMANAC	1425	A collection of all current SBAS almanacs decoded by the receiver
SBASCORR	998	SBAS range corrections used
SOFTLOADSTATUS	1235	Describes the status of the SoftLoad process
SOURCETABLE	1344	Outputs the NTRIP source table entries from the NTRIPCASTER set by the NTRIPSOURCETABLE command
TERRASTARINFO	1719	TerraStar Subscription Information
TERRASTARSTATUS	1729	TerraStar Decoder and Subscription Status
TIME	101	Receiver time information
TIMESYNC	492	Synchronize time between receivers
TRACKSTAT	83	Satellite tracking status
VALIDMODELS	206	Model and expiry date information for receiver
VERIPOSINFO	1728	Veripos Subscription Information
VERIPOSSTATUS	1730	Veripos Decoder and Subscription Status
VERSION	37	Receiver hardware and software version numbers
WIFIAPSTATUS	1666	Wi-Fi Access Point Status
WIFICLISCANRESULTS	1616	Wi-Fi AP scan results
WIFICLISTATUS	1613	Wi-Fi client connection status
CMR Format Logs [a]		
CMRDESC	310	Base station description information
CMRGLOOBS	882	CMR Type 3 GLONASS observations
CMROBS	103	Base station satellite observation information
CMRPLUS	717	Base station position information (low rate)
CMRREF	105	Base station position information
RTCA FORMAT LOGS [a]		
RTCA1	10	Type 1 Differential GPS corrections
RTCAEPHEM	347	Type 7 Ephemeris and time information
RTCAOBS	6	Type 7 Base Station observations
RTCAOBS2	805	Type 7 Base Station observations II
RTCAOBS3	1340	Proprietary message that carries dual-frequency GPS and GLO measurements and is used in ALIGN. Also carries SBAS measurements if the Master receiver is single-frequency (L1-only) receiver to enable SBAS-ALIGN at the L1-only ALIGN Rover

DATA TYPE	MESSAGE ID	DESCRIPTION
RTCAREF	11	Type 7 Base Station parameters
RTCAREFEXT	1049	Type 7 Extended Base Station parameters
RTCM FORMAT LOGS [a]		
RTCM1	107	Type 1 Differential GPS corrections
RTCM3	117	Type 3 Base Station parameters
RTCM9	275	Type 9 Partial Differential GPS corrections
RTCM15	307	Type 15 Ionospheric corrections
RTCM16	129	Type16 Special message
RTCM16T	131	Type16T Special text message
RTCM1819	260	Type18 and Type 19 raw measurements
RTCM2021	374	Type 20 and Type 21 measurement corrections
RTCM22	118	Type 22 Extended Base Station parameters
RTCM23	665	Type 23 Antenna type definition
RTCM24	667	Type 24 Antenna Reference Point (ARP)
RTCM31	864	Type 31 Differential GLONASS corrections
RTCM32	873	Type 32 GLONASS Base Station parameters
RTCM36	875	Type 36 Special message
RTCM36T	877	Type 36T Special text message
RTCM59	116	Type 59N-0 NovAtel Proprietary: RT20 Differential
RTCM59GLO	903	NovAtel proprietary GLONASS differential
RTCMOMNI1	957	RTCM1 from OmniSTAR
RTCMV3 FORMAT LOGS [a]		
RTCM1001	772	L1-Only GPS RTK Observables
RTCM1002	774	Extended L1-Only GPS RTK Observables
RTCM1003	776	L1/L2 GPS RTK Observables
RTCM1004	770	Extended L1/L2 GPS RTK Observables
RTCM1005	765	RTK Base Station ARP
RTCM1006	768	RTK Base Station ARP with Antenna Height
RTCM1007	852	Extended Antenna Descriptor and Setup
RTCM1008	854	Extended Antenna Reference Station Description and serial number
RTCM1009	885	GLONASS L1-Only RTK
RTCM1010	887	Extended GLONASS L1-Only RTK
RTCM1011	889	GLONASS L1/L2 RTK

DATA TYPE	MESSAGE ID	DESCRIPTION
RTCM1012	891	Extended GLONASS L1/L2 RTK
RTCM1019	893	GPS Ephemerides
RTCM1020	895	GLONASS Ephemerides
RTCM1033	1097	Receiver and antenna descriptors
NMEA FORMAT DATA LOGS		
GLMLA	859	NMEA GLONASS almanac data
GPALM	217	Almanac Data
GPGGA	218	GPS Fix Data and undulation
GPGGALONG	521	GPS Fix Data, Extra Precision and undulation
GPGGARTK	259	GPS Fix Data with Extra Precision
GPGLL	219	Geographic Position - latitude/longitude
GPGRS	220	GPS Range Residuals for Each Satellite
GPGSA	221	GPS DOP and Active Satellites
GPGST	222	Pseudorange Measurement Noise Statistics
GPGSV	223	GPS Satellites in View
GPHDT	1045	Heading in Degrees True
GPRMB	224	Generic Navigation Information
GPRMC	225	GPS Specific Information
GPVTG	226	Track Made Good and Ground Speed
GPZDA	227	UTC Time and Date

a.   CMR, RTCA, and RTCM logs may be logged with an A or B extension to give an ASCII or Binary output with a NovAtel header followed by Hex or Binary data respectively

Table 81: OEM6 Logs in Numerical Order

MESSAGE ID	DATA TYPE	DESCRIPTION
5	LOGLIST	A list of system logs
7	GPSEPHEM	GPS ephemeris data
8	IONUTC	Ionospheric and UTC model information
16	CLOCKMODEL	Current clock model matrices
25	RAWGPSSUBFRAME	Raw subframe data
26	CLOCKSTEERING	Clock steering status
37	VERSION	Receiver hardware and software version numbers
41	RAWEPHEM	Raw ephemeris
42	BESTPOS	Best position data

MESSAGE ID	DATA TYPE	DESCRIPTION
43	RANGE	Satellite range information
47	PSRPOS	Pseudorange position information
48	SATVIS	Satellite visibility
72	PORTSTATS	COM or USB port statistics
73	ALMANAC	Current almanac information
74	RAWALM	Raw almanac
83	TRACKSTAT	Satellite tracking status
93	RXSTATUS	Self-test status
94	RXSTATUSEVENT	Status event indicator
96	MATCHEDPOS	RTK Computed Position – Time Matched
99	BESTVEL	Velocity data
100	PSRVEL	Pseudorange velocity information
101	TIME	Receiver time information
128	RXCONFIG	Receiver configuration status
140	RANGECMP	Compressed version of the RANGE log
141	RTKPOS	RTK low latency position data
159	DIRENT	Onboard memory file list
161	NAVIGATE	Navigation waypoint status
172	AVEPOS	Position averaging
174	PSRDOP	DOP of SVs currently tracking
175	REFSTATION	Base station position and health
181	MARKPOS	Position at time of mark1 input event
206	VALIDMODELS	Model and expiry date information for receiver
216	RTKVEL	RTK velocity
231	MARKTIME	Time of mark1 input event
233	PASSCOM1	Pass-through logs
234	PASSCOM2	Pass-through logs
235	PASSCOM3	Pass-through logs
241	BESTXYZ	Cartesian coordinate position data
242	MATCHEDXYZ	RTK Time Matched cartesian coordinate position data
243	PSRXYZ	Pseudorange cartesian coordinate position information
244	RTKXYZ	RTK cartesian coordinate position data
287	RAWSBASFRAME	Raw SBAS frame data

MESSAGE ID	DATA TYPE	DESCRIPTION
405	PASSXCOM1	Pass-through logs
406	PASSXCOM2	Pass-through logs
407	RAWGPSWORD	Raw navigation word
469	PDPPOS	PDP filter position
470	PDPVEL	PDP filter velocity
471	PDPXYZ	PDP filter Cartesian position and velocity
492	TIMESYNC	Synchronize time between receivers
495	OMNIHPPOS	OmniSTAR HP/XP/G2 position data
520	APPLICATIONSTATUS	Provides application status information
607	PASSUSB1	Pass-through logs (for receivers that support USB)
608	PASSUSB2	Pass-through logs (for receivers that support USB)
609	PASSUSB3	Pass-through logs (for receivers that support USB)
615	MARK2POS	Time of mark input2 event
616	MARK2TIME	Position at time of mark2 input event
631	RANGEGPSL1	L1 version of the RANGE log
686	BSLNXYZ	RTK XYZ baseline
690	PASSAUX	Pass-through log for AUX port
718	GLOALMANAC	GLONASS almanac data
719	GLOCLOCK	GLONASS clock information
720	GLORAWALM	Raw GLONASS almanac data
721	GLORAWFRAME	Raw GLONASS frame data
722	GLORAWSTRING	Raw GLONASS string data
723	GLOEPHEMERIS	GLONASS ephemeris data
726	BESTUTM	Best available UTM data
730	LBANDINFO	L-Band configuration information
731	LBANDSTAT	L-Band status information
732	RAWLBANDFRAME	Raw L-Band frame data
733	RAWLBANDPACKET	Raw L-Band data packet
792	GLORAWEPHEM	Raw GLONASS ephemeris data
795	PASSXCOM3	Pass through log
860	OMNIVIS	OmniSTAR satellite visibility list
881	PSRTIME	Time offsets from the pseudorange filter
952	RTKDOP	Values from the RTK fast filter

MESSAGE ID	DATA TYPE	DESCRIPTION
963	HWMONITOR	Monitor Hardware Levels
971	HEADING	Heading information with the ALIGN feature
973	RAWSBASFRAME	Raw SBAS frame data
976	SBAS0	Remove PRN from the solution
977	SBAS1	PRN mask assignments
978	SBAS10	Degradation factor
979	SBAS12	SBAS network time and UTC
980	SBAS17	GEO almanac message
981	SBAS18	IGP mask
982	SBAS2	Fast correction slots 0-12
983	SBAS24	Mixed fast/slow corrections
984	SBAS25	Long term slow satellite corrections
985	SBAS26	Ionospheric delay corrections
986	SBAS27	SBAS service message
987	SBAS3	Fast correction slots 13-25
988	SBAS32	CDGPS Fast Corrections slots 0-10
989	SBAS33	CDGPS Fast Corrections slots 11-21
990	SBAS34	CDGPS Fast Corrections slots 22-32
991	SBAS35	CDGPS Fast Corrections slots 32-43
992	SBAS4	Fast correction slots 26-38
993	SBAS45	CDGPS Slow Corrections
994	SBAS5	Fast corrections slots 39-50
995	SBAS6	Integrity Message
996	SBAS7	Fast Correction Degradation
997	SBAS9	Geo Nav Message
998	SBASCORR	SBAS range corrections used
1201	LBANDTRACKSTAT	L-Band Tracking Status
1043	SATVIS2	Satellite visibility
1051	MASTERPOS	Displays the master position with the ALIGN feature
1052	ROVERPOS	Displays the rover position with the ALIGN feature
1066	RAWCNAVFRAME	Raw L2C frame data
1075	MARK3TIME	Position at time of mark3 input event
1076	MARK4TIME	Position at time of mark4 input event

MESSAGE ID	DATA TYPE	DESCRIPTION
1093	MARK1COUNT	Count for the Mark1 input
1094	MARK2COUNT	Count for the Mark2 input
1095	MARK3COUNT	Count for the Mark3 input
1096	MARK4COUNT	Count for the Mark4 input
1120	GALALMANAC	Decoded Galileo almanac parameters from Galileo navigation messages
1121	GALCLOCK	Galileo time information
1122	GALEPHEMERIS	Galileo ephemeris information is available through the GALEPHEMERIS log
1127	GALIONO	Decoded Galileo ionospheric corrections
1146	LOGFILESTATUS	Current state of file and recording
1148	CHANCONFIGLIST	Channel configuration list
1162	PSRSATS	Satellites used in PSRPOS solution
1163	PSRDOP2	Pseudorange least squares DOP
1172	RTKDOP2	Values from the RTK Fast Filter
1174	RTKSATS	Satellites used in RTKPOS solution
1176	MATCHEDSATS	Lists the used and unused satellites for the corresponding MATCHEDPOS solution
1194	BESTSATS	Satellites used in BESTPOS
1197	OMNIHPSATS	Satellites used in the OMNIHPPOS solution
1209	PASSETH1	Pass through log
1234	PDPSATS	Satellites used in PDPPOS solution
1235	SOFTLOADSTATUS	Describes the status of the SoftLoad process
1250	PASSICOM1	Pass through log
1251	PASSICOM2	Pass through log
1252	PASSICOM3	Pass through log
1253	PASSNCOM1	Pass through log
1254	PASSNCOM2	Pass through log
1255	PASSNCOM3	Pass through log
1273	RANGECMP2	RANGE data compressed to handle more channels and types
1286	RAIMSTATUS	RAIM status
1288	ETHSTATUS	Current Ethernet status
1289	IPSTATUS	Current network configuration status

MESSAGE ID	DATA TYPE	DESCRIPTION
1314	ALIGNBSLNXYZ	Outputs the RTK quality XYZ baselines from ALIGN
1315	ALIGNBSLNENU	Outputs the RTK quality ENU baselines from ALIGN
1316	HEADINGSATS	Outputs the satellite information from ALIGN filter
1325	REFSTATIONINFO	Reference station position and health information
1330	QZSSRAWEPHEM	Contains the raw binary information for subframes one, two and three from the satellite with the parity information removed
1330	QZSSRAWSUBFRAME	A raw QZSS subframe is 300 bits in total, includes the parity bits which are interspersed with the raw data ten times, in six bit chunks, for a total of 60 parity bits
1332	ALIGNDOP	Outputs the DOP computed using the satellites used in solution
1335	HEADING2	Outputs same information as HEADING log with an additional Rover ID field
1336	QZSSEPHEMERIS	Single set of QZSS ephemeris parameters
1340	RTCAOBS3	Proprietary message that carries dual-frequency GPS and GLO measurements and is used in ALIGN. Also carries SBAS measurements if the Master receiver is single-frequency (L1-only) receiver to enable SBAS-ALIGN at the L1-only ALIGN Rover
1342	PASSTHROUGH	Outputs pass-through data from all receiver ports
1344	SOURCETABLE	Outputs the NTRIP source table entries from the NTRIPCASTER set by the NTRIPSOURCETABLE command
1345	QZSSRAWALMANAC	Contains the undecoded almanac subframes as received from the QZSS satellite
1346	QZSSALMANAC	Contains the decoded almanac parameters as received from the satellite with the parity information removed and appropriate scaling applied
1347	QZSSIONUTC	Ionospheric Model parameters (ION) and the Universal Time Coordinated parameters (UTC) for QZSS are provided
1348	AUTHCODES	Contains all authorization codes (auth codes) entered into the system since the last complete firmware reload
1384	PASSCOM4	Pass through log
1412	PROFILEINFO	Outputs a list of Profiles
1413	GALFNAVRAWPAGE	Contains the raw Galileo F/Nav page data
1414	GALINAVRAWWORD	Contains the raw Galileo I/Nav word data
1425	SBASALMANAC	A collection of all current SBAS almanacs decoded by the receiver
1451	SATXYZ2	Combined with a RANGE log, this data set contains the decoded satellite information necessary to compute the solution

MESSAGE ID	DATA TYPE	DESCRIPTION
1538	PPPPOS	PPP filter position
1541	PPPSATS	Satellites used in the PPPPOS solution
1576	PASSCOM5	Pass through log
1577	PASSCOM6	Pass through log
1584	BDSALMANAC	Decoded almanac parameters as received from the satellite, with the parity information removed and appropriate scaling applied
1590	BDSIONO	Contains the Klobuchar ionosphere model parameters transmitted by the BeiDou satellites
1607	BDSCLOCK	Time parameters transmitted by the BeiDou satellites
1608	BLUETOOTHSTATUS	Bluetooth radio module status
1613	WIFICLISTATUS	Wi-Fi client connection status
1616	WIFICLISCANRESULTS	Wi-Fi AP scan results
1618	NOVATELXOBS	NovAtel proprietary RTK correction
1620	NOVATELXREF	NovAtel proprietary reference station message for use in ALIGN
1666	WIFIAPSTATUS	Wi-Fi Access Point Status
1669	IPSTATS	IP statistics
1685	CELLULARSTATUS	Cellular modem and network status information
1686	CELLULARINFO	Cellular modem and network information
1695	BDSRAWNAVSUBFRAME	Log contains single set of BDS ephemeris parameters
1696	BDSEPHEMERIS'	A single set of BDS ephemeris parameters
1698	HEADINGRATE	Provides rate of change for the heading parameters
1701	PASSCOM7	Pass through log (ProPak6 only via expansion cable)
1702	PASSCOM8	Pass through log (ProPak6 only via expansion cable)
1703	PASSCOM9	Pass through log (ProPak6 only via expansion cable)
1704	PASSCOM10	Pass through log (ProPak6 only via expansion cable)
1718	LBANDBEAMTABLE	List of L-Band Beams
1719	TERRASTARINFO	TerraStar Subscription Information
1728	VERIPOSINFO	Veripos Subscription Information
1729	TERRASTARSTATUS	TerraStar Decoder and Subscription Status
1730	VERIPOSSTATUS	Veripos Decoder and Subscription Status
CMR FORMAT LOGS [a]		
103	CMROBS	Base station satellite observation information
105	CMRREF	Base station position information

MESSAGE ID	DATA TYPE	DESCRIPTION
310	CMRDESC	Base station description information
717	CMRPLUS	Base station position information (low rate)
882	CMRGLOOBS	CMR Type 3 GLONASS observations
RTCA FORMAT LOGS [a]		
6	RTCAOBS	Type 7 Base Station observations
10	RTCA1	Type 1 Differential GPS corrections
11	RTCAREF	Type 7 Base Station parameters
347	RTCAEPHEM	Type 7 Ephemeris and time information
805	RTCAOBS2	Type 7 Base Station observations 2
1049	RTCAREFEXT	Type 7 Extended Base Station parameters
RTCM FORMAT LOGS [a]		
107	RTCM1	Type 1 Differential GPS corrections
116	RTCM59	Type 59N-0 NovAtel Proprietary: RT20 differential
117	RTCM3	Type 3 Base Station parameters
118	RTCM22	Type 22 Extended Base Station parameters
129	RTCM16	Type16 Special message
131	RTCM16T	Type16T Special text message
260	RTCM1819	Type18 and Type 19 raw measurements
275	RTCM9	Type 9 Partial Differential GPS Corrections
307	RTCM15	Type 15 Ionospheric Corrections
374	RTCM2021	Type 20 and Type 21 Measurement Corrections
665	RTCM23	Type 22 Extended Base Station parameters
667	RTCM24	Type 23 Antenna Type Definition
864	RTCM31	Type 31 Differential GLONASS Corrections
873	RTCM32	Type 32 GLONASS Base Station parameters
875	RTCM36	Type 36 Special Message
877	RTCM36T	Type 36T Special Text Message
903	RTCM59GLO	NovAtel proprietary GLONASS differential NovAtel proprietary GLONASS differential
957	RTCMOMNI1	RTCM1 from OmniSTAR
RTCMV3 FORMAT LOGS [a]		
765	RTCM1005	RTK Base Station ARP
768	RTCM1006	RTK Base Station ARP with Antenna Height

MESSAGE ID	DATA TYPE	DESCRIPTION
770	RTCM1004	Extended L1/L2 GPS RTK Observables
772	RTCM1001	L1-Only GPS RTK Observables
774	RTCM1002	Extended L1-Only GPS RTK Observables
776	RTCM1003	L1/L2 GPS RTK Observables
852	RTCM1007	Extended Antenna Descriptor and Setup
854	RTCM1008	Extended Antenna Reference Station Description and Serial Number
885	RTCM1009	GLONASS L1-Only RTK
887	RTCM1010	Extended GLONASS L1-Only RTK
889	RTCM1011	GLONASS L1/L2 RTK
891	RTCM1012	Extended GLONASS L1/L2 RTK
893	RTCM1019	GPS Ephemerides
895	RTCM1020	GLONASS Ephemerides
1097	RTCM1033	Receiver and antenna descriptors
NMEA FORMAT DATA LOGS		
217	GPALM	Almanac Data
218	GPGGA	GPS Fix Data and Undulation
219	GPGLL	Geographic Position - latitude/longitude
220	GPGRS	GPS Range Residuals for Each Satellite
221	GPGSA	GPS DOP and Active Satellites
222	GPGST	Pseudorange Measurement Noise Statistics
223	GPGSV	GPS Satellites in View
224	GPRMB	Generic Navigation Information
225	GPRMC	GPS Specific Information
226	GPVTG	Track Made Good and Ground Speed
227	GPZDA	UTC Time and Date
259	GPGGARTK	GPS Fix Data with Extra Precision
521	GPGGALONG	GPS Fix Data, Extra Precision and Undulation
859	GLMLA	NMEA GLONASS Almanac Data
1045	GPHDT	Heading in Degrees True

a. CMR, RTCA, RTCM and RTCMV3 logs may be logged with an A or B extension to give an ASCII or Binary output with a NovAtel header followed by Hex or Binary data respectively.

3.2.1 ALIGNBSLNENU
ENU baselines using ALIGN

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log outputs the RTK quality ENU baselines from ALIGN. The XYZ baselines (output in ALIGNBSLNXYZ log) are rotated relative to master position (output in MASTERPOS) to compute ENU baselines.

Message ID: **1315**
Log Type: **Asynch**

Recommended Input:

> log alignbslnenua onnew

ASCII Example:

```
#ALIGNBSLNENUA,COM1,0,29.0,FINESTEERING,1629,259250.000,00040000,
100B,39448;SOL_COMPUTED,NARROW_INT,4.1586,-1.9197,-0.0037,0.0047,
0.0050,0.0062,"0092","AAAA",22,16,16,16,0,01,0,33*11E1D4C0
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	ALIGNBSLNENU	Log Header		H	0
2	sol stat	Solution status, *see Table 85, Page 408*	Enum	4	H
3	pos type	Position type, *see Table 84, Page 407*	Enum	4	H+4
4	East	East Baseline (relative to master position) in metres	Double	8	H+8
5	North	North Baseline (relative to master position) in metres	Double	8	H+16
6	Up	Up Baseline (relative to master position) in metres	Double	8	H+24
7	East σ	East Baseline STD in metres	Float	4	H+32
8	North σ	North Baseline STD in metres	Float	4	H+36
9	Up σ	Up Baseline STD in metres	Float	4	H+40
10	Rover id	Rover Receiver ID Set using SETROVERID command on ROVER e.g., setroverid RRRR	Char[4]	4	H+44
11	Master id	Master Receiver ID Set using DGPSTXID command on MASTER Default: AAAA	Char[4]	4	H+48
12	#SVs	Number of satellites tracked	Uchar	1	H+52
13	#solnSVs	Number of satellites in solution	Uchar	1	H+53
14	#obs	Number of satellites above elevation mask angle	Uchar	1	H+54

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
15	#multi	Number of satellites above elevation mask angle with L2, B2	Uchar	1	H+55
16	Reserved		Hex	1	H+56
17	ext sol stat	Extended solution status, *see Table 88, Page 410*	Hex	1	H+57
18	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+58
19	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+59
20	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+60
21	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

3.2.2 ALIGNBSLNXYZ
XYZ baselines using ALIGN

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log outputs the RTK quality XYZ baselines from ALIGN.

Message ID: **1314**
Log Type: **Asynch**

Recommended Input:

         log alignbslnxyza onnew

ASCII Example:

```
#ALIGNBSLNXYZA,COM1,0,29.0,FINESTEERING,1629,259250.000,00040000,9D28,
39448;SOL_COMPUTED,NARROW_INT,3.1901,-3.0566, 1.2079,0.0050,0.0054,0.0056,
"0092","AAAA",22,16,16,16,0,01,0,33*AC372198
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	ALIGNBSLNXYZ	Log Header		H	0
2	sol stat	Solution status, see Table 85, Page 408	Enum	4	H
3	pos type	Position type, see Table 84, Page 407	Enum	4	H+4
4	dX	X Baseline in metres	Double	8	H+8
5	dY	Y Baseline in metres	Double	8	H+16
6	dZ	Z Baseline in metres	Double	8	H+24
7	dX σ	X Baseline STD in metres	Float	4	H+32
8	dY σ	Y Baseline STD in metres	Float	4	H+36
9	dZ σ	Z Baseline STD in metres	Float	4	H+40
10	Rover id	Rover Receiver ID Set using SETROVERID command on ROVER e.g. setroverid RRRR	Uchar[4]	4	H+44
11	Master id	Master Receiver Id Set Dgpstxid Command On Master Default: AAAA	Uchar[4]	4	H+48
12	#SVs	Number of satellites tracked	Uchar	1	H+52
13	#solnSVs	Number of satellites in solution	Uchar	1	H+53
14	#obs	Number of satellites above elevation mask angle	Uchar	1	H+54
15	#multi	Number of satellites above elevation mask angle with L2, B2	Uchar	1	H+55
16	Reserved		Hex	1	H+56
17	ext sol stat	Extended solution status, *see Table 88, Page 410*	Hex	1	H+57

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
18	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+58
19	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+59
20	xxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+60
21	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

3.2.3   ALIGNDOP
Calculated DOP values

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log outputs the DOP computed using the satellites used in the heading solution. This log comes out at a default 1 Hz rate. Additional logs may be output not on the even second if the DOP changes and ALIGN is operating at greater than 1 Hz.

Message ID: **1332**
Log Type: **Asynch**

Recommended Input:

　　　log aligndopa onnew

ASCII Example:

```
#ALIGNDOPA,COM1,0,22.5,FINESTEERING,1629,259250.000,00040000,DE2D,
39448;1.6160,1.2400,0.6900,0.9920,0.7130,10.0,16,4,32,23,10,7,20,
13,30,16,47,43,46,53,54,44,45*90A72971
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	ALIGNDOP	Log Header		H	0
2	GDOP	Geometric DOP	Float	4	H
3	PDOP	Position DOP	Float	4	H+4
4	HDOP	Horizontal DOP	Float	4	H+8
5	HTDOP	Horizontal and time DOP	Float	4	H+12
6	TDOP	Time DOP	Float	4	H+16
7	Elev mask	Elevation mask angle	Float	4	H+20
8	#sats	Number of satellites to follow	Ulong	4	H+24
9	sats	Satellites in use at time of calculation	Ulong	4	H+28
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+28+(# sats * 4)
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.4   ALMANAC
Decoded GPS Almanac

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the decoded almanac parameters from subframes four and five, as received from the satellite, with the parity information removed and appropriate scaling applied. For more information about almanac data, refer to the GPS SPS Signal Specification.

The OEM6 family of receivers automatically save almanacs in their Non-Volatile Memory (NVM), so creating an almanac boot file is not necessary.

Message ID: **73**
Log Type: **Asynch**

Recommended Input:

>   log almanaca onchanged

ASCII Example:
```
#ALMANACA,COM1,0,54.0,SATTIME,1364,409278.000,00000000,06DE,2310;
29,
1,1364,589824.0,6.289482E-03,-7.55460039E-09,-2.2193421E+00,
-1.7064776E+00,-7.94268362E-01,4.00543213E-05,3.63797881E-12,
1.45856541E-04,2.6560037E+07,4.45154034E-02,1,0,0,FALSE,
2,1364,589824.0,9.173393E-03,-8.16033991E-09,1.9308788E+00,
1.9904300E+00,6.60915023E-01,-1.62124634E-05,0.00000000,
1.45860023E-04,2.6559614E+07,8.38895743E-03,1,0,0,FALSE,
3,1364,589824.0,7.894993E-03,-8.04604944E-09,7.95206128E-01,
6.63875501E-01,-2.00526792E-01,7.91549683E-05,3.63797881E-12,
1.45858655E-04,2.6559780E+07,-1.59210428E-02,1,0,0,TRUE,
...
28,1364,589824.0,1.113367E-02,-7.87461372E-09,-1.44364969E-01,
-2.2781989E+00,1.6546425E+00,3.24249268E-05,0.00000000,
1.45859775E-04,2.6559644E+07,1.80122900E-02,1,0,0,FALSE,
29,1364,589824.0,9.435177E-03,-7.57745849E-09,-2.2673888E+00,
-9.56729511E-01,1.1791713E+00,5.51223755E-04,1.09139364E-11,
1.45855297E-04,2.6560188E+07,4.36225787E-02,1,0,0,FALSE,
30,1364,589824.0,8.776665E-03,-8.09176563E-09,-1.97082451E-01,
1.2960786E+00,2.0072936E+00,2.76565552E-05,0.00000000,
"1.45849410E-04,2.6560903E+07,2.14517626E-03,1,0,0,FALSE*DE7A4E45
```

> ⓘ The speed at which the receiver locates and locks onto new satellites is improved if the receiver has approximate time and position, as well as an almanac. This allows the receiver to compute the elevation of each satellite so it can tell which satellites are visible and their Doppler offsets, improving Time to First Fix (TTFF).

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	ALMANAC	Log header		H	0
2	#messages	The number of satellite PRN almanac messages to follow. Set to zero until almanac data is available	Long	4	H
3	PRN	Satellite PRN number for current message, dimensionless	Ulong	4	H+4
4	week	Almanac reference week (GPS reference week number)	Ulong	4	H+8
5	seconds	Almanac reference time, seconds into the week	Double	8	H+12
6	ecc	Eccentricity, dimensionless - defined for a conic section where e = 0 is a circle, e = 1 is a parabola, 0<e<1 is an ellipse and e>1 is a hyperbola	Double	8	H+20
7	$\overset{\circ}{\omega}$	Rate of right ascension, radians/second	Double	8	H+28
8	ω_0	Right ascension, radians	Double	8	H+36
9	ω	Argument of perigee, radians - measurement along the orbital path from the ascending node to the point where the SV is closest to the Earth, in the direction of the SV's motion	Double	8	H+44
10	Mo	Mean anomaly of reference time, radians	Double	8	H+52
11	a_{fo}	Clock aging parameter, seconds	Double	8	H+60
12	a_{f1}	Clock aging parameter, seconds/second	Double	8	H+68
13	N	Corrected mean motion, radians/second	Double	8	H+76
14	A	Semi-major axis, metres	Double	8	H+84
15	incl-angle	Angle of inclination relative to 0.3 π, radians	Double	8	H+92
16	SV config	Satellite configuration	Ulong	4	H+100
17	health-prn	SV health from Page 25 of subframe 4 or 5 (6 bits)	Ulong	4	H+104
18	health-alm	SV health from almanac (8 bits)	Ulong	4	H+108
19	antispoof	Anti-spoofing on? 0 = FALSE 1 = TRUE	Bool	4	H+112
20...	Next PRN offset = H + 4 + (#messages x 112)				
21	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H + 4 + (112 x #messages)
22	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.5 APPLICATIONSTATUS
Application status information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains API status information.

Message ID: **520**
Log Type: **Asynch**

Recommended Input:

> log applicationstatus once

ASCII Example:

```
#APPLICATIONSTATUSA,COM2,0,86.0,UNKNOWN,0,1.693,00040000,3314,6096;
1,FALSE,00000000,00000000,"INDIVIDUALTEST","1.00""2010/SEP/14",
"09:31:08"*E3667131
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	APPLICATION STATUS header	Log header		H	0
2	APIVersion	Version of the API header	Ulong	4	H
3	Running	TRUE if the application is running. FALSE otherwise	Bool	4	H+4
4	BaseAddress	Base address of the loaded application	Ulong	4	H+8
5	Size	Size of the loaded application	Ulong	4	H+12
6	Name	Name of the loaded application	FixedCharArray[16]	16	H+16
7	Version	Version of the loaded application	FixedCharArray[16]	16	H+32
8	CompileDate	Compile data of the loaded application	FixedCharArray[12]	12	H+48
9	CompileTime	Compile time of the loaded application	FixedCharArray[12]	12	H+60
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.6 AUTHCODES
List of authorization codes

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains all authorization codes (auth codes) entered into the system since the last complete firmware reload. Signature authorization codes will be maintained through a SoftLoad. The log also indicates the status of the firmware signature, if present. For more information about firmware signatures see the "Upgrading Using the AUTH Command" section of the OEM6 Installation and Operation Manual (OM-20000128).

> The following situations will cause an authorization code to be marked invalid:
> - Authorization Code is for a different receiver
> - Authorization Code is for a different firmware version (if unsigned)
> - Authorization Code has expired
> - Authorization Code was entered incorrectly
> - Authorization Code requires a firmware signature, but one is not present.
>
> If you require new authorization codes, please contact NovAtel Customer Service.

Message ID: 1348
Log Type: Polled

Recommended Input:

         log authcodesa once

ASCII Example:

```
#AUTHCODESA,COM1,0,80.5,UNKNOWN,0,10.775,004C0000,2AD2,12143;
VALID,2,SIGNATURE,TRUE,"63F3K8,MX43GD,T4BJ2X,924RRB,BZRWBT,
D2SB0G550",STANDARD,TRUE,"CJ43M9,2RNDBH,F3PDK8,N88F44,8JMKK9,
D2SB0G550"*6F778E32
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	AUTHCODES header	Log header		H	0
2	AUTHCODES Signature Status	Status of the Firmware Signature 1 = NONE 2 = INVALID 3 = VALID	Enum	4	H
3	Number of Auth Codes	# of Auth Codes to follow (max is 24)	Ulong	4	H+4
4	Auth code type	1=STANDARD 2=SIGNATURE	Enum	4	H+8
5	Valid	TRUE if the Auth Code has been verified	Bool	4	H+12
6	Auth Code String	ASCII String of the Auth Code	String [max 80]	variable[a]	H+16
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+8+ (#AuthCodes*variable)
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

3.2.7   AVEPOS
Position averaging

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

When position averaging is underway, the various fields in the AVEPOS log contain the parameters being used in the position averaging process. *Table 82* below shows the possible position averaging status values seen in field #8 of the AVEPOS log table on the next page.

See the description of the POSAVE command on *page 231.* For general positioning information, refer to our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

> ⓘ   1.   Asynchronous logs should only be logged ONCHANGED. Otherwise, the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

Message ID: **172**
Log Type: **Asynch**

Recommended Input:

    log aveposa onchanged

ASCII Example:

```
#AVEPOSA,COM1,0,48.5,FINESTEERING,1364,492100.000,80000000,E3B4,
2310;51.11635589900,-114.03833558937,1062.216134356,1.7561,0.7856,
1.7236,INPROGRESS,2400,2*72A550C1
```

Table 82: Position Averaging Status

Binary	ASCII	Description
0	OFF	Receiver is not averaging
1	INPROGRESS	Averaging is in progress
2	COMPLETE	Averaging is complete

> (i) When a GNSS position is computed, there are four unknowns being solved: latitude, longitude, height and receiver clock offset (often just called time). The solutions for each of the four unknowns are correlated to satellite positions in a complex way. Since satellites are above the antenna (none are below) there is a geometric bias. Therefore, geometric biases are present in the solutions and affect the computation of height. These biases are called DOPs (Dilution Of Precision). Smaller biases are indicated by low DOP values. VDOP (Vertical DOP) pertains to height. Most of the time, VDOP is higher than HDOP (Horizontal DOP) and TDOP (Time DOP). Therefore, of the four unknowns, height is the most difficult to solve. Many GNSS receivers output the Standard Deviations (SD) of the latitude, longitude and height. Height often has a larger value than the other two.
>
> Accuracy is based on statistics and reliability is measured in percent. When a receiver states it can measure height to one metre, this is an accuracy. Usually this is a one sigma value (one SD). A one sigma value for height has a reliability of 68%. In other words, the error is less than one metre 68% of the time. For a more realistic accuracy, double the one sigma value (one metre) and the result is 95% reliability (error is less than two metres 95% of the time). Generally, GNSS heights are 1.5 times poorer than horizontal positions. See also *page 487* for CEP and RMS definitions.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	AVEPOS header	Log header		H	0
2	lat	Average WGS84 latitude (degrees)	Double	8	H
3	lon	Average WGS84 longitude (degrees)	Double	8	H+8
4	hgt	Average height above sea level (m)	Double	8	H+16
5	lat σ	Estimated average standard deviation of latitude solution element (m)	Float	4	H+24
6	lon σ	Estimated average standard deviation of longitude solution element (m)	Float	4	H+28
7	hgt σ	Estimated average standard deviation of height solution element (m)	Float	4	H+32

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
8	posave	Position averaging status (see *Table 82*)	Enum	4	H+36
9	ave time	Elapsed time of averaging (s)	Ulong	4	H+40
10	#samples	Number of samples in the average	Ulong	4	H+44
11	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+48
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.8 BDSALMANAC
Decoded BDS Almanac

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the decoded almanac parameters, with the parity information removed and appropriate scaling applied. Multiple messages are transmitted, one for each SV almanac collected. For more information about almanac data, refer to the BDS Signal Specification.

The OEM6 family of receivers automatically save almanacs in their Non-Volatile Memory (NVM), so creating an almanac boot file is not necessary.

Message ID: **1584**

Log Type: **Asynch**

Recommended Input:

        log bdsalmanaca onchanged

ASCII Example:

```
#BDSALMANACA,COM1,13,88.5,SATTIME,
1727,518438.000,00000000,24AD,44226;1,371,245760,6493.394531,2.9134
750366E-04,-2.289514637,-0.021819903,-2.456844003,1.30291141E-
09,2.7785425443E-02,-1.096725E-04,2.18279E-11,0*77017E1B
...
#BDSALMANACA,COM1,0,88.5,SATTIME,
1727,518108.000,00000000,24AD,44226;14,371,217088,5282.558105,1.448
6312866E-03,-2.970093901,2.846651891,1.512957087,-6.91457373E-
09,1.7820542434E-02,7.438660E-05,0.00000,D8*CE944672
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	BDSALMANAC header	Log header		H	0
2	satellite ID	Satellite ID/ranging code	Ulong	4	H
3	week	Week number	Ulong	4	H+4
4	toa	Time of almanac	Ulong	4	H+8
5	RootA	Square root of semi-major axis (sqrt(metres))	Double	8	H+12
6	ecc	Eccentricity	Double	8	H+20
7	ω	Argument of perigee (radians)	Double	8	H+28
8	M_0	Mean anomaly at reference time (radians)	Double	8	H+36
9	Ω	Longitude of ascending node of orbital of plane computed according to reference time (radians)	Double	8	H+44

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
10	$\overset{o}{\Omega}$	Rate of right ascension (radians/second)	Double	8	H+52
11	δ_i	Correction of orbit reference inclination at reference time (radians)	Double	8	H+60
12	a_0	Constant term of clock correction polynomial (seconds)	Double	8	H+68
13	a_1	Linear term of clock correction polynomial (seconds/seconds)	Double	8	H+76
14	health	Satellite health information	Ulong	4	H+84
15	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+88
16	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.9   BDSCLOCK
BeiDou time parameters

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains time parameters transmitted by the BeiDou satellites. These parameters can be used to calculated the offset between BeiDou time (BDT) and other time frames.

Message ID: 1607

Log Type: Asynch

Recommended Input:

> log bdsclocka onchanged

ASCII Example:

```
#BDSCLOCKA,COM1,0,80.0,SATTIME,
1730,193994.000,00000000,3B16,44290;-9.313225746154785E-010,-
8.881784197001252E-016,
2,6,0,2,0.000000000000000E+000,0.000000000000000E+000,0.00000000000
0000E+000,0.000000000000000E+000,0.000000000000000E+000,0.000000000
000000E+000*84820676
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	BDSCLOCK header	Log header		H	0
2	A_{0UTC}	BDT clock bias relative to UTC (seconds)	Double	8	H
3	A_{1UTC}	BDT clock rate relative to UTC (seconds/second)	Double	8	H+8
4	ΔT_{LS}	Delta time due to leap seconds before the new leap second is effective (seconds)	Short	2	H+16
5	WN_{LSF}	Week number of the new leap second	Ushort	2	H+18
6	DN	Day number of week of the new leap second	Ushort	2	H+20
7	ΔT_{LSF}	Delta time due to leap seconds after the new leap second effective	Short	2	H+22
8	A_{0GPS}	BDT clock bias relative to GPS time (seconds)	Double	8	H+24
9	A_{1GPS}	BDT clock rate relative to GPS time (seconds/second)	Double	8	H+32
10	A_{0Gal}	BDT clock bias relative to Galileo time (seconds)	Double	8	H+40
11	A_{1Gal}	BDT clock rate relative to Galileo time (seconds/second)	Double	8	H+48
12	A_{0GLO}	BDT clock bias relative to GLONASS time (seconds)	Double	8	H+56
13	A_{1GLO}	BDT clock rate relative to GLONASS time (seconds/second)	Double	8	H+64

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
14	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+68
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.10 BDSEPHEMERIS
Decoded BDS ephemeris

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains a single set of BDS ephemeris parameters with appropriate scaling applied. Multiple messages are transmitted, one for each SV ephemeris collected.

Message ID: 1696

Log Type: Asynch

Recommended Input:

    log bdsephemerisa onchanged

ASCII Example:

```
#BDSEPHEMERISA,COM1,0,82.5,SATTIME,
1774,162464.000,00000000,2626,45436;13,418,2.00,1,8.20E-09,3.10E-
09,11,162000,2.33372441E-04,5.73052716E-12,8.53809211E-
19,12,162000,5282.609060,2.3558507673E-
03,3.122599126,4.1744595973E-09,-0.654635278,1.950232658E+00,-
6.98564812E-09,9.5674299203E-01,3.164417525E-10,4.325527698E-
06,8.850824088E-06,179.3593750,87.5312500,7.171183825E-
08,1.024454832E-08*D8B97536
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	BDSEPHEMERIS header	Log header		H	0
2	satellite ID	ID/ranging code	Ulong	4	H
3	Week	Week number	Ulong	4	H+4
4	URA	User range accuracy (metres). This is the evaluated URAI/URA lookup-table value.	Double	8	H+8
5	health 1	Autonomous satellite health flag. 0 means broadcasting satellite is good and 1 means not.	Ulong	4	H+16
6	tgd1	Equipment group delay differential for the B1 signal (seconds)	Double	8	H+20
7	tgd2	Equipment group delay differential for the B2 signal (seconds)	Double	8	H+28
8	AODC	Age of data, clock	Ulong	4	H+36
9	toc	Reference time of clock parameters	Ulong	4	H+40
10	a_0	Constant term of clock correction polynomial (seconds)	Double	8	H+44

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
11	a_1	Linear term of clock correction polynomial (seconds/seconds)	Double	8	H+52
12	a_2	Quadratic term of clock correction polynomial (seconds/seconds^2)	Double	8	H+60
13	AODE	Age of data, ephemeris	Ulong	4	H+68
14	toe	Reference time of ephemeris parameters	Ulong	4	H+72
15	RootA	Square root of semi-major axis (sqrt(metres))	Double	8	H+76
16	ecc	Eccentricity (sqrt(metres))	Double	8	H+84
17	ω	Argument of perigee	Double	8	H+92
18	ΔN	Mean motion difference from computed value (radians/second)	Double	8	H+100
19	M_0	Mean anomaly at reference time (radians)	Double	8	H+108
20	Ω_0	Longitude of ascending node of orbital of plane computed according to reference time (radians)	Double	8	H+116
21	$\dot{\Omega}$	Rate of right ascension (radians/second)	Double	8	H+124
22	i_0	Inclination angle at reference time (radians)	Double	8	H+132
23	IDOT	Rate of inclination angle (radians/second)	Double	8	H+140
24	c_{uc}	Amplitude of cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+148
25	c_{us}	Amplitude of sine harmonic correction term to the argument of latitude (radians)	Double	8	H+156
26	c_{rc}	Amplitude of cosine harmonic correction term to the orbit radius (metres)	Double	8	H+164
27	c_{rs}	Amplitude of sine harmonic correction term to the orbit radius (metres)	Double	8	H+172

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
28	c_{ic}	Amplitude of cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+180
29	c_{is}	Amplitude of sine harmonic correction term to the angle of inclination (radians)	Double	8	H+188
30	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+196
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.11 BDSIONO
BeiDou Klobuchar ionosphere delay model

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the Klobuchar ionosphere model parameters transmitted by the BeiDou satellites.

Message ID: 1590

Log Type: Asynch

Recommended Input:

       log bdsionoa onchanged

ASCII Example:

```
#BDSIONOA,COM1,0,80.0,SATTIME,
1734,58094.000,00080000,1956,44836;6,2.607703208923340E-
008,4.097819328308105E-007,-3.695487976074218E-
006,7.212162017822263E-006,69632.0,360448.0,-524288.0,-
327680.0*69C2A6C6
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	BDSIONO Header	Log header		H	0
2	ID	Transmitting satellite ID	Ulong	4	H
3	α0	Klobuchar cosine curve amplitude constant term (seconds)	Double	8	H+4
4	α1	Klobuchar cosine curve amplitude first-order term (seconds/π)	Double	8	H+12
5	α2	Klobuchar cosine curve amplitude second-order term (seconds/π2)	Double	8	H+20
6	α3	Klobuchar cosine curve amplitude third-order term (seconds/π3)	Double	8	H+28
7	β0	Klobuchar cosine curve period constant term (seconds)	Double	8	H+36
8	β1	Klobuchar cosine curve period first-order term (seconds/π)	Double	8	H+44
9	β2	Klobuchar cosine curve period second-order term (seconds/π2)	Double	8	H+52
10	β3	Klobuchar cosine curve period third-order term (seconds/π3)	Double	8	H+60
11	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+68
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.12 BDSRAWNAVSUBFRAME
Raw BDS subframe data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw BDS subframe data with parity bits removed. Only subframes that have passed the parity check are output.

Message ID: **1695**

Log Type: **Asynch**

Recommended Input:

      log bdsrawnavsubframea onchanged

ASCII Example:

```
#BDSRAWNAVSUBFRAMEA,COM1,0,85.5,SATTIME,1774,162554.000,00000000,88
F3,45436;84,13,B1D1,1,E24049EBB2B00D113C685207C4D0EE9FD1BF364E41F8F
4B57003268C*6B1F478B
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	BDSRAWNAVSUB FRAME header	Log header		H	0
2	signal channel	Signal channel number	Ulong	4	H
3	satellite ID	Satellite ID	Ulong	4	H+4
4	data source	Source of data (refer to *Table 83, Data Source* on *page 405*)	Enum	4	H+8
5	subframe ID	Subframe identifier	Ulong	4	H+12
6	raw subframe data	Framed raw navigation bits	Hex[28]	28	H+16
7	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 83: Data Source

ASCII	Binary	Description
B1D1	0	Data is from a B1/D1 signal
B1D2	1	Data is from a B1/D2 signal
B2D1	65536	Data is from a B2/D1 signal
B2D2	65537	Data is from a B2/D2 signal

3.2.13   BESTPOS
Best position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

When positioning with GNSS, there are four parameters being solved for: latitude, longitude, height and receiver clock offset from GPS time. The quality of the solution for all four parameters depends on the geometry of where the satellites are with respect to the antenna (and receiver). The strength of the positioning geometry is indicated by Dilution of Precision (DOP) values, with lower DOP numbers indicating better geometry. Because all the GNSS satellites are above terrestrial receivers, the VDOP (vertical DOP) is the largest DOP value. This is why the reported standard deviation for height is usually larger than for latitude or longitude.

Accuracy is based on statistics and reliability is measured in percentages. When a receiver states it can measure height to one metre, this is an accuracy measure. Usually this is a one sigma value (one SD). A one sigma value for height has a reliability of 68%. In other words, the error is less than one metre 68% of the time. For a more realistic accuracy, double the one sigma value (one metre) and the result is 95% reliability (error is less than two metres 95% of the time). Generally, GNSS heights are 1.5 times poorer than horizontal positions. See also the note in the *GPGST Log on page 487* f or CEP and RMS definitions.

This log contains the best position 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, BESTPOS reflects the latest low-latency solution for up to 60 seconds after reception of the last base station observation. 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 RTK mode, pseudorange differential solutions continue for the time specified in the `PSRDIFFTIMEOUT` command. If the receiver is SPAN enabled, the GNSS/INS combined solution is also a candidate for BESTPOS output.

See also the Log by Function table footnotes for position logs as well as the MATCHEDPOS (*page 540*), PSRPOS (*page 581*) and RTKPOS (*page 649*) logs in this manual.

Message ID: **42**
Log Type: **Synch**

Recommended Input:

>   log bestposa ontime 1

> ⓘ   Different positioning modes have different maximum logging rates, which are also controlled by model option. The maximum rates are: 100 Hz for RT-2 (RTK), 100 Hz for pseudorange based positioning, 20 Hz for GLIDE (PDP), 20 Hz for PPP and 20 Hz for OmniSTAR (HP/XP/G2).

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,"",0.000,0.000,8,8,8,8,0,0,0,
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
```

> ⓘ Multi-frequency GNSS receivers offer two major advantages over single-frequency equipment:
>
> 1. Ionospheric errors, inherent in all GNSS observations, can be modelled and significantly reduced by combining satellite observations made on two different frequencies.
> 2. Observations on two frequencies allow for faster ambiguity resolution times.
>
> In general, multi-frequency GNSS receivers provide a faster, more accurate and more reliable solution than single-frequency equipment. They do, however, cost significantly more and so it is important for potential GNSS buyers to carefully consider their current and future needs.

Table 84: Position or Velocity Type

Type (binary)	Type (ASCII)	Description
0	NONE	No solution
1	FIXEDPOS	Position has been fixed by the FIX POSITION command
2	FIXEDHEIGHT	Position has been fixed by the FIX HEIGHT/AUTO command
8	DOPPLER_VELOCITY	Velocity computed using instantaneous Doppler
16	SINGLE	Single point position
17	PSRDIFF	Pseudorange differential solution
18	WAAS	Solution calculated using corrections from an WAAS
19	PROPAGATED	Propagated by a Kalman filter without new observations
20	OMNISTAR [a]	OmniSTAR VBS position
32	L1_FLOAT	Floating L1 ambiguity solution
33	IONOFREE_FLOAT	Floating ionospheric-free ambiguity solution
34	NARROW_FLOAT	Floating narrow-lane ambiguity solution
48	L1_INT	Integer L1 ambiguity solution
50	NARROW_INT	Integer narrow-lane ambiguity solution
64	OMNISTAR_HP [a]	OmniSTAR HP position
65	OMNISTAR_XP [a]	OmniSTAR XP or G2 position
68	PPP_CONVERGING	Converging PPP solution
69	PPP	Converged PPP solution

Type (binary)	Type (ASCII)	Description
70	OPERATIONAL	Solution accuracy is within UAL operational limit
71	WARNING	Solution accuracy is outside UAL operational limit but within warning limit
72	OUT_OF_BOUNDS	Solution accuracy is outside UAL limits

a. In addition to a NovAtel receiver with L-Band capability, a subscription for OmniSTAR or use of a DGPS service is required. Contact NovAtel for details. NovAtel CORRECT™ with PPP requires access to a suitable correction stream, delivered either through L-Band or the internet. For L-Band delivered TerraStar or Veripos service, an L-Band capable receiver and software model is required, along with a subscription to the desired service. Contact NovAtel for TerraStar and Veripos subscription details.

ⓘ If SPAN enabled, refer the SPAN on SPAN on OEM6 Firmware Reference Manual (OM-20000144) located on our web site: www.novatel.com/support/.

Table 85: Solution Status

Solution Status (Binary)	Solution Status (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 export licensing restrictions)
8	VARIANCE	Variance exceeds limits
9	RESIDUALS	Residuals are too large
13	INTEGRITY_WARNING	Large residuals make position unreliable
18	PENDING	When a `FIX POSITION` command is entered, the receiver computes its own position and determines if the fixed position is valid [a]
19	INVALID_FIX	The fixed position, entered using the `FIX POSITION` command, is not valid
20	UNAUTHORIZED	Position type is unauthorized - HP or XP on a receiver not authorized for it

a. PENDING implies there are not enough satellites currently tracked to verify if the FIX POSITION entered into the receiver is valid. Under normal conditions, you should only see PENDING for a few seconds on power up before the GNSS receiver has locked onto its first few satellites. If your antenna is obstructed (or not plugged in) and you have entered a FIX POSITION command, then you may see PENDING indefinitely.

Table 86: BESTPOS 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 87: BESTPOS 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 88: Extended Solution Status

Bit	Mask	Description
0	0x01	If an RTK solution: NovAtel CORRECT solution has been verified If a PDP solution: solution is GLIDE Otherwise: Reserved

1-3	0x0E	Pseudorange Iono Correction
		      0 = Unknownn or default Klobuchar model
		      1 = Klobuchar Broadcast
		      2 = SBAS Broadcast
		      3 = Multi-frequency Computed
		      4 = PSRDiff Correction
		      5 = NovAtel Blended Iono Value

4	0x10	Reserved
5	0x20	0 - No antenna warning 1 - Antenna information is missing See to the RTKANTENNA command
6-7	0xC0	Reserved

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	BESTPOS header	Log header		H	0
2	sol stat	Solution status, see *Table 85* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84* on *page 407*	Enum	4	H+4
4	lat	Latitude 9 (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 23, Reference Ellipsoid Constants* on *page 119*)	Enum	4	H+36
9	lat σ	Latitude standard deviation (m)	Float	4	H+40
10	lon σ	Longitude standard deviation (m)	Float	4	H+44
11	hgt σ	Height standard deviation (m)	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 satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites used in solution	Uchar	1	H+65

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
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		Hex	1	H+68
20	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	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 89: Supplemental Position Types and NMEA Equivalents

Value	Documented Enum Name	NMEA Equivalent
68	PPP_CONVERGING	2
69	PPP	5
70	OPERATIONAL	4
71	WARNING	5
72	OUT_OF_BOUNDS	1

3.2.14   BESTSATS
Satellites used in BESTPOS

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log lists the used and unused satellites for the corresponding BESTPOS solution. It also describes the signals of the used satellites or reasons for exclusions.

Message ID: **1194**
Log Type: **Synch**

Recommended Input:

>   log bestsats ontime 1

Abbreviated ASCII Example:

```
<BESTSATS COM1 0 57.5 FINESTEERING 1729 12132.000 00000000 95E7 11487
<     26
<         GPS 3 GOOD 00000003
<         GPS 5 GOOD 00000003
...
<         GPS 26 GOOD 00000003
<         GPS 28 GOOD 00000003
<         GLONASS 3+5 GOOD 00000003
<         GLONASS 4+6 GOOD 00000003
...
<         GLONASS 23+3 GOOD 00000003
<         GLONASS 24+2 GOOD 00000003
<         BEIDOU 6 GOOD 00000003
<         BEIDOU 9 GOOD 00000003
...
<         BEIDOU 12 GOOD 00000003
<         BEIDOU 13 GOOD 00000003
```

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	BESTSATS header	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	System	See *Table 110* on *page 508*	Enum	4	H+4

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
4	Satellite ID	In binary logs, the satellite ID field is 4 bytes. The 2 lowest-order bytes, interpreted as a USHORT, are the system identifier: for instance, the PRN for GPS, or the slot for GLONASS. The 2 highest-order bytes are the frequency channel for GLONASS, interpreted as a SHORT and zero for all other systems. In ASCII and abbreviated ASCII logs, the satellite ID field is the system identifier. If the system is GLONASS and the frequency channel is not zero, then the signed channel is appended to the system identifier. For example, slot 13, frequency channel -2 is output as 13-2	Ulong	4	H+8
5	Status	Satellite status. See *Table 90*	Enum	4	H+12
6	Signal mask	See *Table 91*, *Table 92*, *Table 93* and *Table 94*	Hex	4	H+16
7	Next satellite offset = H + 4 + (# x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4 (# sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 90: Observation Statuses

Value	Name	Description
0	GOOD	Observation is good
1	BADHEALTH	Satellite is flagged as bad health in ephemeris or almanac
2	OLDEPHEMERIS	Ephemeris >3 hours old
6	ELEVATIONERROR	Satellite was below the elevation cutoff
7	MISCLOSURE	Observation was too far from predicted value
8	NODIFFCORR	No differential correction available
9	NOEPHEMERIS	No ephemeris available
10	INVALIDIODE	IODE used is invalid
11	LOCKEDOUT	Satellite has been locked out
12	LOWPOWER	Satellite has low signal power
13	OBSL2	An L2 observation not directly used in the solution
15	UNKNOWN	Observation was not used because it was of an unknown type
16	NOIONOCORR	No ionosphere delay correction was available
17	NOTUSED	Observation was not used in the solution
18	OBSL1	An L1 observation not directly used in the solution
19	OBSE1	An E1 observation not directly used in the solution

Value	Name	Description
20	OBSL5	An L5 observation not directly used in the solution
21	OBSE5	An E5 observation not directly used in the solution
22	OBSB2	A B2 observation not directly used in the solution
23	OBSB1	A B1 observation not directly used in the solution
25	NOSIGNALMATCH	Signal type does not match
26	SUPPLEMENTARY	Observation contributes supplemental information to the solution
99	NA	No observation available
100	BAD_INTEGRITY	Observation was an outlier and was eliminated from the solution
101	LOSSOFLOCK	Lock was broken on this signal
102	NOAMBIGUITY	No RTK ambiguity type resolved

Table 91: BESTSATS GPS Signal 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

Table 92: BESTSATS GLONASS Signal Mask

Bit	Mask	Description
0	0x01	GLONASS L1 used in Solution
1	0x02	GLONASS L2 used in Solution

Table 93: BESTSATS Galileo Signal Mask

Bit	Mask	Description
0	0x01	Galileo E1 used in Solution

Table 94: BESTSATS BeiDou Signal Mask

Bit	Mask	Description
0	0x01	BeiDou B1 used in Solution
1	0x02	BeiDou B2 used in Solution

3.2.15 BESTUTM
Best available UTM data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the best available position computed by the receiver in UTM coordinates.

See also the UTMZONE command on *page 347* and the BESTPOS log on *page 406*.

Message ID: **726**
Log Type: **Synch**

> (i) The latitude limits of the UTM System are 80°S to 84°N. If your position is outside this range, the BESTUTM log outputs a northing, easting and height of 0.0, along with a zone letter of '*'and a zone number of 0, to indicate that the data in the log is unusable.

Recommended Input:

log bestutma ontime 1

ASCII Example:

```
#BESTUTMA,COM1,0,73.0,FINESTEERING,1419,336209.000,00000040,EB16,
2724;SOL_COMPUTED,NARROW_INT,11,U,5666936.4417,707279.3875,
1063.8401,-16.2712,WGS84,0.0135,0.0084,0.0173,"AAAA",1.000,0.000,
8,8,8,8,0,01,0,03*A6D06321
```

> (i) Refer to http://earth-info.nga.mil/GandG/coordsys/grids/referencesys.html for more information and a world map of UTM zone numbers.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	BESTUTM header	Log header		H	0
2	sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+4
4	z#	Longitudinal zone number	Ulong	4	H+8
5	zletter	Latitudinal zone letter	Ulong	4	H+12
6	northing	Northing (m) where the origin is defined as the equator in the northern hemisphere and as a point 10000000 metres south of the equator in the southern hemisphere (that is, a 'false northing' of 10000000 m)	Double	8	H+16

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
7	easting	Easting (m) where the origin is 500000 m west of the central meridian of each longitudinal zone (that is, a 'false easting' of 500000 m)	Double	8	H+24
8	hgt	Height above mean sea level (m)	Double	8	H+32
9	undulation	Undulation - the relationship between the geoid and the ellipsoid (m) of the chosen datum [a]	Float	4	H+40
10	datum id#	Datum ID number (see *Table 23, Reference Ellipsoid Constants* on *page 119*)	Enum	4	H+44
11	N σ	Northing standard deviation (m)	Float	4	H+48
12	E σ	Easting standard deviation (m)	Float	4	H+52
13	hgt σ	Height standard deviation (m)	Float	4	H+56
14	stn id	Base station ID	Char[4]	4	H+60
15	diff_age	Differential age in seconds	Float	4	H+64
16	sol_age	Solution age in seconds	Float	4	H+68
17	#SVs	Number of satellites tracked	Uchar	1	H+72
18	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+73
19	#ggL1	Number of GPS plus GLONASS plus BDS L1/B1 used in solution	Uchar	1	H+74
20	#solnMultiSV	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+75
21	Reserved		Uchar	1	H+76
22	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+77
23	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+78
24	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+79
25	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+80
26	[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.

3.2.16   BESTVEL
Best available velocity data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the best available velocity information 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 have a latency associated with them. The time of validity is the time tag in the log minus the latency value. See also the table footnote for velocity logs on *page 362*.

> (i)   The velocity is typically from the same source used in the BESTPOS solution. For example, if the BESTPOS is from the pseudorange filter, then the BESTVEL velocity type is the same as for PSRVEL. However, a specific velocity source can be chosen. See the `BESTLEVTYPE` command on page 97.

The RTK PPP, OmniSTAR XP/HP/G2 velocities are computed from the average change in position over the time interval between consecutive solutions. As such, it is an average velocity based on the time difference between successive position computations and not an instantaneous velocity at the BESTVEL 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 average velocity translates into a velocity latency of 0.25 seconds.

To reduce the latency, increase the update rate of the positioning filter being used by requesting the BESTVEL or BESTPOS messages at a rate higher than 2 Hz. For example, a logging rate of 10 Hz would reduce the velocity latency to 0.05 seconds. For integration purposes, the velocity latency should be applied to the record time tag.

If the velocity in the BESTVEL log comes from the pseudorange filter, it has been computed from instantaneous doppler measurements. You know that you have an instantaneous doppler velocity solution when you see PSRDIFF, WAAS, OMNISTAR or DOPPLER_VELOCITY in field #3 (*vel type*). The instantaneous doppler velocity has low latency and is not delta position dependent. If you change your velocity quickly, you can see this in the DOPPLER_VELOCITY solution. This instantaneous doppler velocity translates into a velocity latency of 0.15 seconds.

Message ID: **99**
Log Type: **Synch**

Recommended Input:

> log bestvela ontime 1

ASCII Example:
```
#BESTVELA,COM1,0,61.0,FINESTEERING,1337,334167.000,00000000,827B,
1984;SOL_COMPUTED,PSRDIFF,0.250,4.000,0.0206,227.712486,0.0493,
0.0*0E68BF05
```

> (i) Velocity (speed and direction) calculations are computed from either Doppler or carrier phase measurements rather than from pseudorange measurements. Typical speed accuracies are around 0.03m/s (0.07 mph, 0.06 knots).
>
> Direction accuracy is derived as a function of the vehicle speed. A simple approach would be to assume a worst case 0.03 m/s cross-track velocity that would yield a direction error function something like:
>
> d (speed) = $\tan^{-1}$(0.03/speed)
>
> For example, if you are flying in an airplane at a speed of 120 knots or 62 m/s, the approximate directional error will be:
>
> $\tan^{-1}$ (0.03/62) = 0.03 degrees
>
> Consider another example applicable to hiking at an average walking speed of 3 knots or 1.5 m/s. Using the same error function yields a direction error of about 1.15 degrees.
>
> You can see from both examples that a faster vehicle speed allows for a more accurate heading indication. As the vehicle slows down, the velocity information becomes less and less accurate. If the vehicle is stopped, a GNSS receiver still outputs some kind of movement at speeds between 0 and 0.5 m/s in random and changing directions. This represents the noise and error of the static position.
>
> In a navigation capacity, the velocity information provided by your GNSS receiver is as, or more, accurate than that indicated by conventional instruments as long as the vehicle is moving at a reasonable rate of speed. It is important to set the GNSS measurement rate fast enough to keep up with all major changes of the vehicle's speed and direction. It is important to keep in mind that although the velocity vector is quite accurate in terms of heading and speed, the actual track of the vehicle might be skewed or offset from the true track by plus or minus 0 to 1.8 metres as per the standard positional errors.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	BESTVEL header	Log header		H	0
2	sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	vel type	Velocity type, see *Table 84, Position or Velocity Type* on *page 407*	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 (s)	Float	4	H+8
5	age	Differential age in seconds	Float	4	H+12
6	hor spd	Horizontal speed over ground, in metres per second	Double	8	H+16

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
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)	-	-	-

3.2.17 *BESTXYZ*
Best available cartesian position and velocity

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the receiver's best available position and velocity in ECEF coordinates. The position and velocity status fields indicate whether or not the corresponding data is valid. See *Figure 11, The WGS84 ECEF Coordinate System* on *page 422,* for a definition of the ECEF coordinates.

See also the BESTPOS (*page 406*) and BESTVEL (*page 417*) logs.

> (i) These quantities are always referenced to the WGS84 ellipsoid, regardless of the use of the DATUM or USERDATUM commands.

Message ID: **241**
Log Type: **Synch**

Recommended Input:

        log bestxyza ontime 1

ASCII Example:

```
#BESTXYZA,COM1,0,55.0,FINESTEERING,1419,340033.000,00000040,D821,
2724;SOL_COMPUTED,NARROW_INT,-1634531.5683,-3664618.0326,
4942496.3270,0.0099,0.0219,0.0115,SOL_COMPUTED,NARROW_INT,0.0011,
-0.0049,-0.0001,0.0199,0.0439,0.0230,"AAAA",0.250,1.000,0.000,
12,11,11,11,0,01,0,33*E9EAFECA
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	BESTXYZ header	Log header		H	0
2	P-sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X σ	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y σ	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z σ	Standard deviation of P-Z (m)	Float	4	H+40
10	V-sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H+44

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
11	vel type	Velocity type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+48
12	V-X	Velocity vector along X-axis (m/s)	Double	8	H+52
13	V-Y	Velocity vector along Y-axis (m/s)	Double	8	H+60
14	V-Z	Velocity vector along Z-axis (m/s)	Double	8	H+68
15	V-X σ	Standard deviation of V-X (m/s)	Float	4	H+76
16	V-Y σ	Standard deviation of V-Y (m/s)	Float	4	H+80
17	V-Z σ	Standard deviation of V-Z (m/s)	Float	4	H+84
18	stn ID	Base station identification	Char[4]	4	H+88
19	V-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+92
20	diff_age	Differential age in seconds	Float	4	H+96
21	sol_age	Solution age in seconds	Float	4	H+100
22	#SVs	Number of satellites tracked	Uchar	1	H+104
23	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+105
24	#ggL1	Number of GPS plus GLONASS plus BDS L1/B1 used in solution	Uchar	1	H+106
25	#solnMultiSVs	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+107
26	Reserved		Char	1	H+108
27	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+109
28	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+110
29	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+111
30	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+112
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Figure 11: The WGS84 ECEF Coordinate System
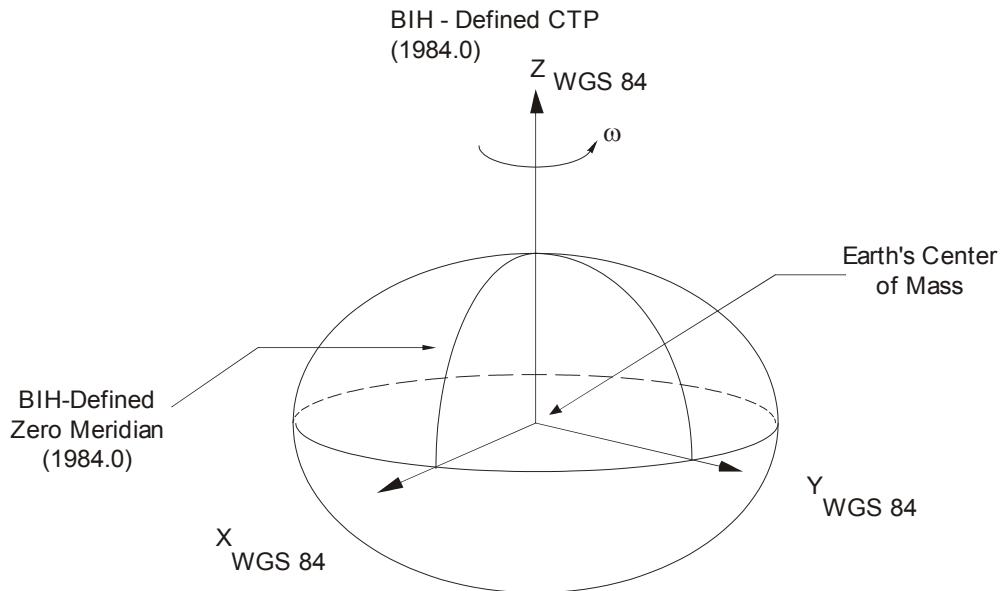
- Definitions - *****

Origin =   Earth's center of mass

Z-Axis =   Parallel to the direction of the Conventional Terrestrial Pole (CTP) for polar motion, as defined by the Bureau International de l'Heure (BIH) on the basis of the coordinates adopted for the BIH stations.

X-Axis =   Intersection of the WGS 84 Reference Meridian Plane and the plane of the CTP's Equator, the Reference Meridian being parallel to the Zero Meridian defined by the BIH on the basis of the coordinates adopted for the BIH stations.

Y-Axis =   Completes a right-handed, earth-centered, earth-fixed (ECEF) orthogonal coordinate system, measured in the plane of the CTP Equator, 90° East of the X-Axis.

*   Analogous to the BIH Defined Conventional Terrestrial System (CTS), or BTS, 1984.0.

3.2.18   BLUETOOTHSTATUS
Bluetooth radio module status

OEM Platform: ProPak6

This log displays the status of the Bluetooth radio module.

Message ID: **1608**
Log Type: **Asynch**

Recommended Input:

       log bluetoothstatusa onchanged

ASCII Example:

```
#BLUETOOTHSTATUSA,COM1,0,65.5,FINESTEERING,1745,420975.939,
00440020,E870,45001;OFF,"","",""*0E0DD582
```

```
#BLUETOOTHSTATUSA,COM1,0,79.0,FINESTEERING,1745,420989.673,
00040020,E870,45001;ON,"PROPAK-6 BMAW13130046Y","B1:9E:65:2F:18:0",
""*D2CE72EB
```

```
#BLUETOOTHSTATUSA,COM1,0,66.5,FINESTEERING,1745,421012.417,00040020
,E870,45001;PASSKEY,"DEL44385","36:D2:BB:72:2:0","859002"*74134842
```

```
#BLUETOOTHSTATUSA,COM1,0,76.5,FINESTEERING,1745,421020.142,00040020
,E870,45001;CONNECTED,"DEL44385","36:D2:BB:72:2:0",""*EBC826D1
```

Field#	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	BLUETOOTH STATUS Header	Log Header			0
2	BluetoothStatus	Current Bluetooth status. See *Table 95, Bluetooth Status* on *page 424*	Enum	4	H
3	Device Name	• Is blank if the Bluetooth module is off • Displays the local device name if no remote device is connected • Displays the remote device name if a remote device is connected	String [Max 272]	Variable[a]	H+4
4	MAC Address	• Is blank if the Bluetooth module is off • Displays the local device MAC address if no remote device is connected • Displays the remote device MAC address if a remote device is connected	String [Max 64])	Variable[a]	Variable Max: H+276
5	Pass Code	• Displays the pass code if pairing with a remote device using SSP • Is blank otherwise	String [Max 64]	Variable[a]	Variable Max: H+340

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 95: Bluetooth Status

Binary	ASCII	Description
0	ON	Bluetooth module is on, but not discoverable. This is the actual state rather than the configured state. Refer to BLUETOOTHCONFIG on *page 98* for the configured state.
1	OFF	Bluetooth module is off. This is the actual state.
2	CONNECTED	Serial Port (BT SPP) is connected.
3	PASSKEY	Secure Simple Pairing passkey displayed for confirmation.
4	DISCOVERABLE	Bluetooth module is on and discoverable, allowing devices to detect the ProPak6 and pair with it.

3.2.19   BSLNXYZ
RTK XYZ baseline

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the receiver's RTK baseline in ECEF coordinates. The position status field indicates whether or not the corresponding data is valid. See *Figure 11, The WGS84 ECEF Coordinate System* on *page 422* for a definition of the ECEF coordinates.

The BSLNXYZ log comes from time-matched base and rover observations such as in the MATCHEDXYZ log on *page 543*.

> (i) Asynchronous logs, such as BSLNXYZ, should only be logged ONCHANGED. Otherwise, the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

Message ID: **686**
Log Type: **Asynch**

Recommended Input:

        log bslnxyza onchanged

ASCII Example:

```
#BSLNXYZA,COM1,0,59.5,FINESTEERING,1419,340033.000,00000040,5B48,
2724;SOL_COMPUTED,NARROW_INT,0.0012,0.0002,-0.0004,0.0080,0.0160,
0.0153,"AAAA",12,12,12,12,0,01,0,33*1A8A1B65.
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	BSLNXYZ header	Log header		H	0
2	sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	bsln type	Baseline type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+4
4	B-X	X-axis offset (m)	Double	8	H+8
5	B-Y	Y-axis offset (m)	Double	8	H+16
6	B-Z	Z-axis offset (m)	Double	8	H+24
7	B-X σ	Standard deviation of B-X (m)	Float	4	H+32
8	B-Y σ	Standard deviation of B-Y (m)	Float	4	H+36
9	B-Z σ	Standard deviation of B-Z (m)	Float	4	H+40
10	stn ID	Base station identification	Char[4]	4	H+44
11	#SVs	Number of satellites tracked	Uchar	1	H+48

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
12	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+49
13	#ggL1	Number of GPS plus GLONASS plus BDS L1/B1 used in solution	Uchar	1	H+50
14	#solnMultiSVs	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+51
15	Reserved		Uchar	1	H+52
16	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+53
17	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+54
18	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+55
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+56
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.20   CELLULARINFO
Cellular modem and network information

OEM Platform: ProPak6

This log displays information about the cellular modem and network settings.

Message ID: **1686**
Log Type: **Asynch**

Recommended Input:

> log cellularinfoa once

ASCII Example:

```
#CELLULARINFOA,COM1,0,72.0,UNKNOWN,0,125.980,014C4020,250D,45068;
"HE910-D","TELIT","351579952407038","12.00.023","",""*2566D58F
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	CELLULARINFO header	Log header		H	0
2	make	Modem make	String [max 20]	Variable[a]	H
3	manufacturer	Modem manufacturer	String [max 20]	Variable[a]	Variable Max: H+20
4	serial	The International Mobile Equipment Identifier (IMEI).	String [max 32]	Variable[a]	Variable Max: H+40
5	version	Modem software version	String [max 64]	Variable[a]	Variable Max: H+72
6	mdn[b]	Mobile Directory Number (MDN), the modem phone number	String [max 16]	Variable[a]	Variable Max: H+136
7	msid[b]	Mobile Subscriber Identifier (MSID)	String [max 32]	Variable[a]	Variable Max: H+152

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

b. The MDN and MSID fields may be blank (empty strings).
   With some carriers, this information cannot be reliably obtained so the fields will not be filled in or will be filled in after some delay.

3.2.21   CELLULARSTATUS
Cellular modem and network status information

OEM Platform: ProPak6

This log displays the current status of the cellular modem and the cellular connection.

Message ID: **1685**
Log Type: **Asynch**

Recommended Input:

> log cellularstatusa onchanged

ASCII Example:

```
#CELLULARSTATUSA,COM1,0,76.0,UNKNOWN,0,1653.887,014C4020,F714,
45068;FULL,REGISTERED_HOME,"196.207.248.183",2,18,"AIRTEL",12D009A,
34,"NORMAL, UNSPECIFIED"*40A40551
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	CELLULARSTATUS header	Log header		H	0
2	status	Current modem status See *Table 96, Modem Status* on *page 429*	Enum	4	H
3	net status	Registration status of the modem on the network. See *Table 97, Network Status* on *page 429*	Enum	4	H+4
4	ip address	Network-assigned IP address	String [Max 16]	Variable[a]	H+8
5	signal	Signal strength as number of bars (1-4)	Long	4	Variable Max: H+24
6	rssi	Received Signal Strength Indicator (dBm)	Long	4	Variable Max: H+28
7	network	Network identification string or NID	String [Max 20]	Variable[a]	Variable Max: H+32
8	cellid	Base station cell identifier	Ulong	4	Variable Max: H+52
9	temperature	Modem temperature, if available. If the temperature is not available, this value is 0.	Long	4	Variable Max: H+56
10	last error	Last recorded modem error	String [Max 100]	Variable[a]	Variable Max: H+60

  a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 96: Modem Status

Modem Status Binary	ASCII	Description
1	UNKNOWN	
2	IN_PROGRESS	Power control is in progress, power state is indeterminate
3	OFF	Radio is powered off
4	SIM_NOT_INSERTED	SIM is not inserted
5	SIM_PASSWORD_REQUIRED	Radio is powered on; SIM password is required; some functionality is unavailable
6	FULL	Radio is fully powered on

Table 97: Network Status

Network Status Binary	ASCII	Description
0	NOTREGISTERED_NOTSEARCHING	Not registered and not currently searching for a new operator to register to.
1	REGISTERED_HOME	Registered on the home network
2	NOTREGISTERED_SEARCHING	Not registered, but currently searching a for new operator to register to
3	DENIED	Registration denied
4	UNKNOWN	Unknown
5	REGISTERED_ROAMING	Registered on network that is not the home network (roaming)

3.2.22 CHANCONFIGLIST
Channel configuration list

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log provides the channel configuration list including the number of channels and signal types. If more than one channel configuration is available, then it can be switched using the SELECTCHANCONFIG command.

Message ID: **1148**

Log Type: **Polled**

Recommended Input:

      log chanconfiglista once

Abbreviated ASCII Example:
```
CHANCONFIGLIST COM1 2 73.5 FINESTEERING 1783 585128.718 01000040 D1C0
12793
4 4
6
12 GPSL1L2PL5
2 QZSSL1CAL2CL5
2 SBASL1
10 GLOL1L2
9 GALE1E5AE5BALTBOC
10 BEIDOUB1B2
6
10 GPSL1L2PL2CL5
2 QZSSL1CAL2CL5
2 SBASL1
8 GLOL1L2PL2C
8 GALE1E5AE5BALTBOC
8 BEIDOUB1B2
6
12 GPSL1L2PL5
2 QZSSL1CAL2CL5
2 SBASL1L5
10 GLOL1L2
9 GALE1E5AE5BALTBOC
9 BEIDOUB1B2
6
9 GPSL1L2PL2CL5
2 QZSSL1CAL2CL5
2 SBASL1L5
8 GLOL1L2PL2C
8 GALE1E5AE5BALTBOC
9 BEIDOUB1B2
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	CHANCONFIGLIST header	Log header		H	0
2	SetInUse	Current channel configuration being used. For example, if SetInUse is 2 then the second channel configuration listed in this log is the current channel configuration	Ulong	4	H
3	#chanconfigs	Number of channel configurations to follow	Ulong	4	H+4
4	#signaltypes	Total number of signal types in this channel configuration	Ulong	4	H+8
5	NumChans	Number of channels for individual signal type	Ulong	4	H+12
6	SignalType	See *Table 98* on *page 432*	Ulong	4	H+16
7	Next chanconfig offset = H + 8+ (#chanconfigs * (4 + (#signaltypes * 8)))					
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 98: CHANCONFIGLIST Signal Type

Value	Name	
0	GPSL1	GPS L1 C/A signal
1	GPSL1L2	GPS L1 C/A and L2P(Y) signal
4	SBASL1	SBAS L1 C/A signal
5	GPSL5	GPS L5 signal
6	GPSL1L2C	GPS L1 C/A and L2C signal
7	GPSL1L2AUTO	GPS L1 C/A and L2 P(Y) or L2C signal
8	GLOL1L2	GLONASS L1 C/A and L2P signal
9	LBAND	L-Band signal
10	GLOL1	GLONASS L1 C/A signal
11	GALE1	Galileo E1 signal
12	GALE5A	Galileo E5a signal
13	GALE5B	Galileo E5b signal
14	GALALTBOC	Galileo E5 AltBOC signal
15	BEIDOUB1	BeiDou B1 signal
16	GPSL1L2PL2C	GPS L1 C/A, L2 P(Y), and L2C signal
17	GPSL1L5	GPS L1 C/A and L5 signal
18	SBASL1L5	SBAS L1 C/A and L5 signal
19	GPSL1L2PL2CL5	GPS L1 C/A, L2 P(Y), L2C, and L5 signal
20	GPSL1L2PL5	GPS L1 C/A, L2 P(Y), and L5 signal
21	GALE1E5AE5B	Galileo E1, E5a, and E5b signal
22	GALE1E5AE5BALTBOC	Galileo E1, E5a, E5b, and E5 AltBOC signal
23	GALE1E5A	Galileo E1 and E5a signal
24	GLOL1L2C	GLONASS L1 C/A and L2C signal
25	GLOL1L2PL2C	GLONASS L1 C/A, L2 P, and L2C signal
26	QZSSL1CA	QZSS L1 C/A signal
27	QZSSL1CAL2C	QZSS L1 C/A and L2C signal
28	QZSSL1CAL2CL5	QZSS L1 C/A, L2C, and L5 signal
29	QZSSL1CAL5	QZSS L1 C/A and L5 signal
30	BEIDOUB1B2	BeiDou B1 and B2 signal
31	GALE1E5B	Galileo E1 and E5b signal

3.2.23 CLOCKMODEL
Current clock model status

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The CLOCKMODEL log contains the current clock model status of the receiver.

Monitoring the CLOCKMODEL log allows you to determine the error in your receiver reference oscillator as compared to the GNSS satellite reference.

All logs report GPS reference time not corrected for local receiver clock error. To derive the closest GPS reference time, subtract the clock offset from the GPS reference time reported. The clock offset can be calculated by dividing the value of the range bias given in field *6* of the CLOCKMODEL log by the speed of light (*c*).

The following symbols are used throughout this section:

B = range bias (m)

BR = range bias rate (m/s)

SAB = Gauss-Markov process representing range bias error due to satellite clock dither (m)

The standard clock model now used is as follows:

clock parameters array = [B   BR   SAB]

covariance matrix =

$$
\begin{bmatrix}
\sigma_B^2 & \sigma_B\sigma_{BR} & \sigma_B\sigma_{SAB} \\
\sigma_{BR}\sigma_B & \sigma_{BR}^2 & \sigma_{BR}\sigma_{SAB} \\
\sigma_{SAB}\sigma_B & \sigma_{SAB}\sigma_{BR} & \sigma_{SAB}^2
\end{bmatrix}
$$

Table 99: Clock Model Status

Clock Status (Binary)	Clock Status (ASCII)	Description
0	VALID	The clock model is valid
1	CONVERGING	The clock model is near validity
2	ITERATING	The clock model is iterating towards validity
3	INVALID	The clock model is not valid
4	ERROR	Clock model error

Message ID: **16**
Log Type: **Synch**

Recommended Input:

       log clockmodela ontime 1

ASCII Example:

```
#CLOCKMODELA,COM1,0,52.0,FINESTEERING,1364,489457.000,80000000,
98F9,2310;VALID,0,489457.000,489457.000,7.11142843E+00,
6.110131956E-03,-4.93391151E+00,3.02626565E+01,2.801659017E-02,
-2.99281529E+01,2.801659017E-02,2.895779736E-02,-1.040643538E-02,
-2.99281529E+01,-1.040643538E-02,3.07428979E+01,2.113,
2.710235665E-02,FALSE*3D530B9A
```

> ⓘ The CLOCKMODEL log can be used to monitor the clock drift of an internal oscillator once the CLOCKADJUST mode has been disabled. Watch the CLOCKMODEL log to see the drift rate and adjust the oscillator until the drift stops.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	CLOCKMODEL header	Log header		H	0
2	clock status	Clock model status as computed from current measurement data, see *Table 99, Clock Model Status* on *page 433*	Enum	4	H
3	reject	Number of rejected range bias measurements	Ulong	4	H+4
4	noise time	GPS reference time of last noise addition	GPSec	4	H+8
5	update time	GPS reference time of last update	GPSec	4	H+12
6	parameters	Clock correction parameters (a 1x3 array of length 3), listed left-to-right	Double	8	H+16
7				8	H+24
8				8	H+32
9	cov data	Covariance of the straight line fit (a 3x3 array of length 9), listed left-to-right by rows	Double	8	H+40
10				8	H+48
11				8	H+56
12				8	H+64
13				8	H+72
14				8	H+80
15				8	H+88
16				8	H+96
17				8	H+104
18	range bias	Last instantaneous measurement of the range bias (metres)	Double	8	H+112
19	range bias rate	Last instantaneous measurement of the range bias rate (m/s)	Double	8	H+120

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
20	Reserved		Bool	4	H+128
21	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+132
22	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.24   CLOCKSTEERING
Clock steering status

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The CLOCKSTEERING log is used to monitor the current state of the clock steering process. All oscillators have some inherent drift. By default the receiver attempts to steer the receiver's clock to accurately match GPS reference time. If for some reason this is not desired, this behavior can be disabled using the CLOCKADJUST command, see *page 102*.

> (i) If the CLOCKADJUST command is ENABLED and the receiver is configured to use an external reference frequency (set in the EXTERNALCLOCK command, see *page 151*, then the clock steering process takes over the VARF output pins and may conflict with a previously entered FREQUENCYOUT command, see *page 161*.

Message ID: **26**
Log Type: **Asynch**

Recommended Input:

      log clocksteeringa onchanged

ASCII Example:

```
#CLOCKSTEERINGA,COM1,0,56.5,FINESTEERING,1337,394857.051,00000000,
0F61,1984;INTERNAL,SECOND_ORDER,4400,1707.554687500,0.029999999,
-2.000000000,-0.224,0.060*0E218BBC
```

> (i) To configure the receiver to use an external reference oscillator, see the EXTERNALCLOCK command on *page 151.*

Table 100: Clock Source

Binary	ASCII	Description
0	INTERNAL	The receiver is currently steering its internal VCTCXO using an internal VARF signal
1	EXTERNAL	The receiver is currently steering an external oscillator using the external VARF signal

Table 101: Steering State

Binary	ASCII	Description
0	FIRST_ORDER	Upon start-up, the clock steering task adjusts the VARF pulse width to reduce the receiver clock drift rate to below 1 ms using a 1st order control loop. This is the normal start-up state of the clock steering loop.
1	SECOND_ORDER	Once the receiver has reduced the clock drift to below 1 m/s, it enters a second order control loop and attempts to reduce the receiver clock offset to zero. This is the normal runtime state of the clock steering process.

Binary	ASCII	Description
2	CALIBRATE_HIGH [a]	This state corresponds to when the calibration process is measuring at the "High" pulse width setting.
3	CALIBRATE_LOW [a]	This state corresponds to when the calibration process is measuring at the "Low" pulse width setting.
4	CALIBRATE_CENTER [b]	This state corresponds to the "Center" calibration process. Once the center has been found, the modulus pulse width, center pulse width, loop bandwidth and measured slope values are saved in NVM and are used from now on for the currently selected oscillator (INTERNAL or EXTERNAL).

   a. These states are only seen if you force the receiver to do a clock steering calibration using the CLOCKCALIBRATE command, see *page 104*. With the CLOCKCALIBRATE command, you can force the receiver to calibrate the slope and center pulse width of the currently selected oscillator, to steer. The receiver measures the drift rate at several "High" and "Low" pulse width settings.

   b. After the receiver has measured the "High" and "Low" pulse width setting, the calibration process enters a "Center calibration" process where it attempts to find the pulse width required to zero the clock drift rate.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	CLOCKSTEERING header	Log header		H	0
2	source	Clock source, see *Table 100, Clock Source on page 436*	Enum	4	H
3	steeringstate	Steering state, see *Table 101, Steering State on page 436*	Enum	4	H+4
4	period	Period of the FREQUENCYOUT signal used to control the oscillator, refer to the FREQUENCYOUT command. This value is set using the CLOCKCALIBRATE command	Ulong	4	H+8
5	pulsewidth	Current pulse width of the FREQUENCYOUT signal. The starting point for this value is set using the CLOCKCALIBRATE command. The clock steering loop continuously adjusts this value in an attempt to drive the receiver clock offset and drift terms to zero	Double	8	H+12
6	bandwidth	The current band width of the clock steering tracking loop in Hz. This value is set using the CLOCKCALIBRATE command	Double	8	H+20
7	slope	The current clock drift change in m/s/bit for a 1 LSB pulse width. This value is set using the CLOCKCALIBRATE command	Float	4	H+28
8	offset	The last valid receiver clock offset computed (m). It is the same as Field # 18 of the CLOCKMODEL log, see *page 433*	Double	8	H+32

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
9	driftrate	The last valid receiver clock drift rate received (m/s). It is the same as Field # 19 of the CLOCKMODEL log	Double	8	H+40
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+48
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.25 CMR Standard Logs

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

CMRDESC	**BASE Station Description Information**
Message ID:	**310**
CMRGLOOBS	**CMR Data GLONASS Observations (cmr tYPE 3 mESSAGE)**
Message ID:	**882**
CMROBS	**BASE Station Satellite Observation Information**
Message ID:	**103**
CMRPLUS	**BASE STATION POSITION INFORMATION (LOW RATE)**
Message ID:	**717**
CMRREF	**BASE Station Position Information**
Message ID:	**105**

The Compact Measurement Record (CMR) Format is a standard communications protocol used in Real-Time Kinematic (RTK) systems to transfer GNSS carrier phase and code observations from a base station to one or more rover stations.

> (i)
> 1. The above messages can be logged with an A or B suffix for an ASCII or Binary output with a NovAtel header followed by Hex or Binary raw data respectively.
> 2. No guarantee is made that the OEM6 will meet its performance specifications if non-NovAtel equipment is used to provide differential corrections to the OEM6.
> 3. Trimble rovers must receive CMRDESC messages from a base.

The CMR message format was developed by Trimble Navigation Ltd. as a proprietary data transmission standard for use in RTK applications. In 1996, Trimble publicly disclosed this standard and allowed its use by all manufacturers in the GNSS industry[1].

The NovAtel implementation allows a NovAtel rover receiver to operate in RTK mode while receiving pseudorange and carrier phase data via CMR messages (version 3.0) from either a NovAtel or non-NovAtel base-station receiver. The NovAtel receiver can also transmit CMR messages (version 3.0). The station ID must be ≤ 31 when transmitting CMR corrections. The CMRPLUS output message distributes the base station information over 14 updates.

The maximum message lengths of the four CMR messages are as follows:

CMROBS = 6 *(frame)* + 6 *(header)* + (14*L1 channels) + (14*L2 channels) = (222 bytes max.)

CMRREF = 6 *(frame)* + 6 *(header)* + 19 = (31 bytes)

CMRDESC = 6 (*frame*) + 6 (*header*) + (variable: 26 to 75) = (38 bytes minimum; 87 bytes max.)

CMRPLUS = 6 (*frame*) + 3 (*header*) + 7 = (16 bytes)

---

1. Talbot, N.C. (1996) "Compact Data Transmission Standard for High-Precision GPS". ION GPS-96 Conference Proceedings, Kansas, MO, Sept. 1996, Vol. I, pp. 861-871.

CMR Type 3 RTK Formats

> (i)   NovAtel CMR Type 3 messages are CMR Type 3 messages as defined by Leica and Topcon.

CMR Type 3 format messages are for GLONASS CMR observations. The CMRGLOOBS log is similar to the existing CMROBS log. See also *CMR Standard Logs* starting on *page 439.*

CMR Type 3 message types (CMRGLOOBS) have their Z count stamped to GLONASS UTC time instead of GPS reference time (the *epoch* field in the CMR Header part of the message).

When using CMRGLOOBS in conjunction with CMRREF and CMROBS, to perform GPS + GLONASS RTK positioning (provided you have a GLONASS-capable receiver model).

CMR Type 3 Example Setup

In the example below, apply *Steps #1* and *#2* to the base and *Step #3* to the rover.

1. Use the INTERFACEMODE command to set up the base port's receive mode as NONE and transmit mode as CMR:

   ```
   interfacemode com2 none cmr
   ```

2. Log out CMRREF, CMROBS and CMRGLOOBS [1] messages:

   ```
   log com2 CMRREF ontime 10
   log com2 CMROBS ontime 1
   log com2 CMRGLOOBS ontime 1
   log com2 CMRDDESC ontime 10
   ```

> (i)   We recommend that you log CMROBS and CMRGLOOBS messages out at the same rate.

3. Set up the rover receiver to use incoming CMR messages by setting the rover port's receive mode as CMR and the transmit mode as NONE:

   ```
   interfacemode com2 CMR none
   ```

Using AdVance RTK with CMR Format Messages

To enable receiving CMR messages, follow these steps:

1. Issue the SERIALCONFIG command, see *page 282*, to the rover receiver to set its serial port parameters to the proper bit rate parity and so on.

2. Issue the "INTERFACEMODE COMn CMR" command to the rover receiver, where "COMn" refers to the communication port that is connected to the data link. See also *page 184*.

---

1.   These correspond to reference station data, GPS observations and GLONASS observations respectively.

To send CMR messages, periodically transmit the three following CMR messages at the base station:

- A CMROBS message that contains base station satellite observation information. It should be sent once every 1 or 2 seconds.

- A CMRREF message that contains base station position information. It should be sent once every 10 seconds. Also, the rover receiver automatically sets an approximate position from this message if it does not already have a position. Therefore, this message can be used in conjunction with an approximate time to improve Time to First Fix (TTFF). For more information about TTFF, refer to our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

- A CMRDESC message that contains base station description information. It should be sent once every 10 seconds and be interlinked with the CMRREF message.

Assuming that the base station is transmitting valid data, your rover receiver begins to operate in AdVance RTK mode.

ⓘ
1. For CMR, the station ID must be less than 31 (refer to the DGPSTXID and RTKSOURCE commands on *pages 125* and *266* respectively).

2. CMRDESC is logged with an offset of 5 to allow interleaving with CMRREF. Note that Trimble rovers must receive CMRDESC messages from a base.

3. NovAtel CMR Type 2 messages are for compatibility only. Type 2 provides a reference station description message that is required by some manufacturers' rovers before CMR messages are used. When received, a Type 2 message is discarded. For transmission, all fields are permanently set as follows:

    Record Length=33 bytes
    Short Station ID = "cref"
    COGO Code   = ""
    Long Station ID="UNKNOWN"

Example Input:

interfacemode com2 none cmr

fix position 51.116372360734 -114.038308797   1047.575

log com2 cmrobs ontime 1

log com2 cmrref ontime 10

log com2 cmrdesc ontime 10 5

log com2 cmrgloobs ontime 1

3.2.26 COMCONFIG

> (i) COMCONFIG is replaced with the SERIALCONFIG command (*page 282*) which can act as a log when parameters are not defined.

3.2.27   DIRENT
Onboard memory file list

OEM Platform: 638, ProPak6

The DIRENT log contains the current file contents of the receiver's onboard memory. Up to 512 files can be listed using this message.

The date and time for the DIRENT log is in UTC (Universal Coordinated Time). If the receiver has no almanac, UTC is unavailable.

The Date of Last Change field has 4 decimal digits reserved for the year, followed by 2 decimal digits reserved for the month, and 2 decimal digits for the day.

Example: Date of September 5, 2013

Date of Last Change field for this date when converted to a Ulong has a value of 20130905. The following steps can be used to obtain the various parts of the Date of Last Change field:

Ulong Year = (Ulong)(DateOfLastChange / 10000)

Ulong Month = (Ulong)((DateOfLastChange - (Year * 10000)) / 100)

Ulong Day = (Ulong)(DateOfLastChange - (Year * 10000) - (Month * 100))

The Time of Last Change field is similar to the Date of Last Change field, in that the value of the field has 2 decimal digits reserved for the hour, followed by 2 decimal digits for the minutes, and 2 decimal digits for the seconds.

Example: Time of 16:01:25

Time of Last Change field when converted to a Ulong has a value of 160125. The following steps can be used to obtain the various parts of the Time of Last Change field:

Ulong Hour = (Ulong)(TimeOfLastChange / 10000)

Ulong Minutes = (Ulong)((TimeOfLastChange - (Hour * 10000)) / 100)

Ulong Seconds = (Ulong)(TimeOfLastChange - (Hour * 10000) - (Minutes * 100))

Message ID = 159
Log Type: Polled

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	DIRENT Header	Log header	-		0
2	Filename	File name	Char[ ]	128	H
3	Sizebytes	Bytes size (bytes)	Ulong	4	H+128
4	Sizepackets	Packet size (packets)	Ulong	4	H+132
5	Lastchangedate	Date of last change (yyyymmdd)	Ulong	4	H+136
6	Lastchangetime	Time of last change (hhmmss)	Ulong	4	H+140
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.28 ETHSTATUS
Current Ethernet status

OEM Platform: 628, 638, FlexPak6, ProPak6

This log provides the current status of the Ethernet ports.

Message ID: **1288**
Log Type: **Async**

Recommended Input:

        log ethstatusa once

ASCII Example:

```
#ETHSTATUSA,COM1,0,89.5,FINESTEERING,1609,500138.174,00000000,E89D,
6259;1,ETHA,"00-21-66-00-05-A2",100_FULL*98D86B04
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	ETHSTATUS Header	Log header	-	H	0
2	#of interfaces	Number of records to follow	Ulong	4	H
3	interface	Name of the Ethernet interface (e.g., ETHA)	Enum	4	H+4
4	MAC address	An identifier assigned to the network adapters or network interface card	String [18]	variable[a]	H+8
5	interface configuration	Current connectivity, speed and duplex settings of the Ethernet interface	Enum	4	H+26
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+(# of interfaces * 26)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

> ⓘ   Refer to the ETHCONFIG command (*page 146*) for enum values.

3.2.29   GALALMANAC
Decoded Galileo Almanac

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the decoded Galileo almanac parameters from Galileo navigation messages. Multiple messages are transmitted, one for each SVID with data

This log is not finalized and is subject to change.

Message ID: 1120
Log Type: Asynch

Recommended Input:

> log galalmanaca onchanged

ASCII Example:

```
#GALALMANACA,COM1,3,83.5,SAT-
TIME,1769,333371.000,00000020,131F,45362;19,FALSE,TRUE,0,0,0,0,10,7
45,332400.000,1.221E-04,-5.486E-09,2.757E+00,2.038E+00,-1.226E+00,-
1.1444E-05,0.000,2.539E-02,-1.457E-02*5C77F44B
```

```
#GALALMANACA,COM1,2,83.5,SAT-
TIME,1769,333399.000,00000020,131F,45362;20,FALSE,TRUE,0,0,0,0,10,7
45,332400.000,1.831E-04,-5.486E-09,2.757E+00,1.542E+00,-3.1734E-
02,4.8084E-03,9.495E-10,2.539E-02,-1.457E-02*3530E391
```

```
#GALALMANACA,COM1,1,83.5,SAT-
TIME,1769,333939.000,00000020,131F,45362;11,FALSE,TRUE,0,0,0,0,11,7
45,333000.000,6.104E-05,-5.120E-09,6.6412E-01,2.396E+00,-
1.032E+00,5.1498E-05,1.091E-11,3.125E-02,-1.764E-02*AFA0F631
```

```
#GALALMANACA,COM1,0,83.5,SAT-
TIME,1769,333941.000,00000020,131F,45362;12,FALSE,TRUE,0,0,0,0,11,7
45,333000.000,1.526E-04,-5.120E-09,6.6412E-01,-2.392E+00,-
1.818E+00,6.4850E-05,1.091E-11,3.516E-02,-1.764E-02*EF41E1B2
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	GALALMANAC header	Log header		H	0
2	SatId	Satellite ID	Ulong	4	H
3	FNAVReceived	Indicates FNAV almanac data received	Bool	4	H+4
4	INAVReceived	Indicates INAV almanac data received	Bool	4	H+8
5	E1BHealth	E1B health status bits (only valid if INAVReceived is TRUE)	Uchar	1	H+12
6	E5aHealth	E5a health status bits (only valid if FNAVReceived is TRUE)	Uchar	1	H+13

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
7	E5bHealth	E5b health status bits (only valid if INAVReceived is TRUE)	Uchar	1	H+14
8	Reserved		Uchar	1	H+15
9	IODa	Almanac issue of data	Ulong	4	H+16
10	Weeks	Almanac reference week, modulus 1024	Ulong	4	H+20
11	Seconds	Almanac reference time of week (seconds for ASCII, milliseconds for binary)	GPSec	4	H+24
12	Ecc	Eccentricity	Double	8	H+28
13	OmegaDot	Rate of right ascension	Double	8	H+36
14	Omega0	Right ascension	Double	8	H+44
15	Omega	Argument of perigee	Double	8	H+52
16	M0	Mean anomaly at ref time	Double	8	H+60
17	Af0	Satellite clock correction bias	Double	8	H+68
18	Af1	Satellite clock correction linear	Double	8	H+76
19	DeltaRootA	Difference with respect to the square root of the nominal semi-major axis	Double	8	H+84
20	DeltaI	Inclination at reference time relative to I0 = 56 deg	Double	8	H+92
21	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+100
22	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.30 GALCLOCK
Galileo clock information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the Galileo time information.

> ▼ This log is not finalized and is subject to change.

> ⓘ This log is populated from both the INAV and FNAV messages. Depending on the data source, it is possible that the time in the header of the log may be earlier than the time in a previous log. This is expected behavior.

Message ID: 1121
Log Type: Asynch

Recommended Input:

    log galclocka onchanged

ASCII Example:

```
#GALCLOCKA,COM1,0,84.5,SATTIME,
1769,336845.000,00000020,C6CF,45362;8.381903172E-09,-3.5527137E-
15,16,259200,233,28,7,16,-3.5216E-09,-1.776E-14,345600,41*186E9085
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	GALCLOCK header	Log header		H	0
2	A0	Constant term of polynomial	Double	8	H
3	A1	1st order term of polynomial	Double	8	H+8
4	DeltaTls	Leap second count before leap second adjustment	Long	4	H+16
5	Tot	UTC data reference time of week (seconds)	Ulong	4	H+20
6	WNt	UTC data reference week number	Ulong	4	H+24
7	WNlsf	Week number of leap second adjustment	Ulong	4	H+28
8	DN	Day number at the end of which a leap second adjustment becomes effective	Ulong	4	H+32
9	DeltaTlsf	Leap second count after leap second adjustment	Long	4	H+36
10	A0g	Constant term of the polynomial describing the difference between Galileo and GPS time	Double	8	H+40
11	A1g	Rate of change of offset the offset between Galileo and GPS time	Double	8	H+48
12	T0g	Reference time for GGTO datas	Ulong	4	H+56
13	WN0g	Week number of GGTO reference	Ulong	4	H+60

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
14	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+64
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.31 GALEPHEMERIS
Decoded Galileo Ephemeris

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains Galileo ephemeris information.Multiple messages are transmitted, one for each SVID with date.

▼ This log is not finalized and is subject to change.

ⓘ This log is populated from both the INAV and FNAV messages. Depending on the data source, it is possible that the time in the header of the log may be earlier than the time in a previous log. This is expected behavior.

Message ID: 1122
Log Type: Asynch

Recommended Input:

log galephemerisa onchanged

ASCII Example:

```
#GALEPHEMERISA,COM1,3,84.0,SATTIME,1769,336895.000,00000020,DDC6,45
362;20,FALSE,TRUE,0,0,0,0,0,0,107,0,48,336000,5.44061421E+03,3.3351
E-09,4.135230286E-01,1.833668211E-04,1.54330383E+00,-5.1595E-
07,1.0144E-05,1.245E+02,-1.206E+01,3.7253E-09,6.8918E-
08,9.628509837E-01,-7.5646E-10,2.75683149E+00,-5.4634419E-
09,0,0.000000000,0.000000,0.0,336000,4.811083607E-03,9.504078E-
10,0.0,3.260E-09,3.492E-09*4A101918
```

```
#GALEPHEMERISA,COM1,2,84.0,SATTIME,1769,336055.000,00000020,DDC6,45
362;12,FALSE,TRUE,0,0,0,0,0,0,107,0,46,334800,5.44062124E+03,2.7433
E-09,-1.59939066E+00,1.471719006E-04,-2.38667040E+00,-4.3102E-
06,1.4253E-05,2.994E+01,-9.513E+01,2.6077E-08,4.8429E-
08,9.596726435E-01,6.8789E-10,6.640948000E-01,-5.2684337E-
09,0,0.000000000,0.000000,0.0,334800,6.539805327E-05,1.249134E-
11,0.0,-2.328E-09,-2.095E-09*06A29C83
```

```
#GALEPHEMERISA,COM1,1,84.0,SATTIME,1769,336655.000,00000020,DDC6,45
362;19,FALSE,TRUE,0,0,0,0,0,0,107,0,47,335400,5.44061331E+03,3.3259
E-09,-8.547636369E-01,1.281467266E-04,2.03894226E+00,-4.9546E-
07,9.9950E-06,1.260E+02,-1.138E+01,9.8720E-08,0.0000,9.628405311E-
01,-8.1611E-10,2.75686609E+00,-5.4988005E-
09,0,0.000000000,0.000000,0.0,335400,-1.169519965E-05,-4.831691E-
13,0.0,3.958E-09,4.657E-09*6AE89727
```

```
#GALEPHEMERISA,COM1,0,84.0,SATTIME,1769,336895.000,00000020,DDC6,45
362;11,FALSE,TRUE,0,0,0,0,0,0,107,0,48,336000,5.44062407E+03,2.7673
E-09,-6.419769592E-01,6.538478192E-05,2.37749875E+00,-4.5095E-
06,1.4078E-05,3.494E+01,-9.647E+01,2.6077E-08,2.2352E-
08,9.596617345E-01,6.5467E-10,6.641122644E-01,-5.2645050E-
09,0,0.000000000,0.000000,0.0,336000,5.154800601E-05,1.030287E-
11,0.0,-4.889E-09,-6.054E-09*B19BAEF3
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	GALEPHEMERIS header	Log header		H	0
2	SatId	Satellite ID	Ulong	4	H
3	FNAVReceived	Indicates FNAV ephemeris data received	Bool	4	H+4
4	INAVReceived	Indicates INAV ephemeris data received	Bool	4	H+8
5	E1BHealth	E1B health status bits (only valid if INAVReceived is TRUE)	Uchar	1	H+12
6	E5aHealth	E5a health status bits (only valid if FNAVReceived is TRUE)	Uchar	1	H+13
7	E5bHealth	E5b health status bits (only valid if INAVReceived is TRUE)	Uchar	1	H+14
8	E1BDVS	E1B data validity status (only valid if INAVReceived is TRUE)	Uchar	1	H+15
9	E5aDVS	E5a data validity status (only valid if FNAVReceived is TRUE)	Uchar	1	H+16
10	E5bDVS	E5b data validity status (only valid if INAVReceived is TRUE)	Uchar	1	H+17
11	SISA	Signal in space accuracy	Uchar	1	H+18
12	Reserved		Uchar	1	H+19
13	IODNav	Issue of data ephemeris	Ulong	4	H+20
14	T0e	Ephemeris reference time (s)	Ulong	4	H+24
15	RootA	Square root of semi-major axis	Double	8	H+28
16	DeltaN	Mean motion difference (radians/s)	Double	8	H+36
17	M0	Mean anomaly at ref time (radians)	Double	8	H+44
18	Ecc	Eccentricity (unitless)	Double	8	H+52
19	Omega	Argument of perigee (radians)	Double	8	H+60
20	Cuc	Amplitude of the cosine harmonic correction term to the argument of latitude (radians)	Double	8	H+68
21	Cus	Amplitude of the sine harmonic correction term to the argument of latitude (radians)	Double	8	H+76

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
22	Crc	Amplitude of the cosine harmonic correction term to the orbit radius (m)	Double	8	H+84
23	Crs	Amplitude of the sine harmonic correction term to the orbit radius (m)	Double	8	H+92
24	Cic	Amplitude of the cosine harmonic correction term to the angle of inclination (radians)	Double	8	H+100
25	Cis	Amplitude of the sine harmonic correction term to the angle of inclination (radians)	Double	8	H+108
26	I0	Inclination angle at ref time (radians)	Double	8	H+116
27	IDot	Rate of inclination angle (radians/s)	Double	8	H+124
28	Omega0	Longitude of ascending node of orbital plane at weekly epoch (radians)	Double	8	H+132
29	OmegaDot	Rate of right ascension (radians/s)	Double	8	H+140
30	FNAVT0c	Clock correction data reference time of week from the F/NAV message (s). Only valid if FNAVReceived is TRUE	Ulong	4	H+148
31	FNAVAf0	SV clock bias correction coefficient from the F/NAV message (s). Only valid if FNAVReceived is TRUE	Double	8	H+152
32	FNAVAf1	SV clock drift correction coefficient from the F/NAV message (s/s). Only valid if FNAVReceived is TRUE	Double	8	H+160
33	FNAVAf2	SV clock drift rate correction coefficient from the F/NAV message (s/s^2). Only valid if FNAVReceived is TRUE	Double	8	H+168
34	INAVT0c	Clock correction data reference time of week from the I/NAV message (s). Only valid if INAVReceived is TRUE	Ulong	4	H+176
35	INAVAf0	SV clock bias correction coefficient from the I/NAV message (s). Only valid if INAVReceived is TRUE	Double	8	H+180
36	INAVAf1	SV clock drift correction coefficient from the I/NAV message (s/s). Only valid if INAVReceived is TRUE	Double	8	H+188
37	INAVAf2	SV clock drift rate correction coefficient from the I/NAV message (s/s^2). Only valid if INAVReceived is TRUE	Double	8	H+196
38	E1E5aBGD	E1, E5a broadcast group delay	Double	8	H+204
39	E1E5bBGD	E1, E5b broadcast group delay. Only valid if INAVReceived is TRUE	Double	8	H+212
40	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+220
41	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.32 GALFNAVRAWPAGE
Raw Galileo F/NAV page data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw Galileo F/Nav page data.

▼   This log is not finalized and is subject to change.

Message ID: 1413
Log Type: Asynch

Recommended Input:

> log galfnavrawpagea onchanged

ASCII Example:

```
#GALFNAVRAWPAGEA,USB3,0,85.0,SATTIME,1680,434410.000,00000008,D4FB,
43274;56,11,0B818DF50AD5FFC151001BAFFDAA04D5DAE655E17AF-
FC8A41A83AA*5955B14D
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	GALFNAVRAWPAGE header	Log header		H	0
2	SigChanNum	Signal channel providing the data	Ulong	4	H
3	SatId	SVID of transmitting satellite	Ulong	4	H+4
4	RawFrameData	Raw F/NAV page (214 bits). Does not include CRC or Tail bits	Hex[27]	27	H+8
5	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+35
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.33 GALINAVRAWWORD
Raw Galileo I/NAV word data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw Galileo I/Nav word data.

> ▼ This log is not finalized and is subject to change.

Message ID: 1414
Log Type: Asynch

Recommended Input:

log galinavrawworda onchanged

ASCII Example:
```
#GALINAVRAWWORDA,USB3,0,84.5,SATTIME,1680,434401.000,00000008,884B,
43274;55,11,GALE1,0B81E655E17A26EB5237D7D20088FFC9*DCB4BEDB
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	GALINAVRAWWORD header	Log header		H	0
2	SigChanNum	Signal channel providing data	Ulong	4	H
3	SatId	SVID of transmitting satellite	Ulong	4	H+4
4	SignalType	Signal Type as defined in *Table 25, Signal Type* on *page 130*	Enum	4	H+8
5	RawFrameData	Raw I/NAV word (128 bits)	Hex[16]	16	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+28
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.34 GALIONO
Decoded Galileo ionospheric corrections

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the decoded Galileo ionospheric corrections.

> ▼ This log is not finalized and is subject to change.

Message ID: 1127
Log Type: Asynch

Recommended Input:

> log galionoa onchanged

ASCII Example:
```
#GALIONOA,COM1,0,84.0,SATTIME,1686,503485.000,00000020,D22E,10636;
100.000000000,0.000000000,0.000000000,0,0,0,0,0*5215B367
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	GALIONO header	Log header		H	0
2	Ai0	Effective ionisation level 1st order parameter (sfu)	Double	8	H
3	Ai1	Effective ionisation level 2st order parameter (sfu/degree)	Double	8	H+8
4	Ai2	Effective ionisation level 3st order parameter (sfu/degree^2)	Double	8	H+16
5	SF1	Ionospheric disturbance flag for region 1	Uchar	1	H+24
6	SF2	Ionospheric disturbance flag for region 2	Uchar	1	H+25
7	SF3	Ionospheric disturbance flag for region 3	Uchar	1	H+26
8	SF4	Ionospheric disturbance for flag region 4	Uchar	1	H+27
9	SF5	Ionospheric disturbance for flag region 5	Uchar	1	H+28
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+29
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.35 GLMLA
NMEA GLONASS Almanac data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log outputs almanac data for GLONASS satellites. Multiple sentences are transmitted, one for each satellite.

> (i) The following relationships enable translation between the NMEA GLONASS satellite IDs, the NovAtel GLONASS PRN IDs, and the GLONASS slot numbers:
> NMEA GLONASS satellite ID = GLONASS slot number + 64
> NovAtel GLONASS PRN ID   = GLONASS slot number + 37
>   = NMEA GLONASS satellite ID - 27

Message ID: 859
Log Type: Asynch

Recommended Input:

    log glmlaa onchanged

ASCII Example:

```
$GLMLA,16,01,65,1176,07,0496,4C,5FF2,8000,34C05E,0E93E8,04B029,
001FA2,099,213*68
$GLMLA,16,02,66,1176,01,12E3,4C,42CC,8000,34C08E,10FAE9,02F48C,
00224E,099,003*64
$GLMLA,16,03,67,1176,8C,08F6,4A,EF4D,8000,34C051,13897B,00D063,
001B09,099,000*63
$GLMLA,16,04,68,1176,06,116B,48,3A00,8000,34C09D,02151F,0E49E8,
00226E,099,222*63
$GLMLA,16,05,70,1176,01,140F,49,45C4,8000,34C0BC,076637,0A3E40,
002214,099,036*37
$GLMLA,16,06,71,1176,05,0306,4C,5133,8000,34C025,09BDA7,085D84,
001F83,099,21D*6E
$GLMLA,16,07,72,1176,06,01B1,4C,4C19,8000,34C021,0C35A0,067DB8,
001FCA,099,047*3D
$GLMLA,16,08,74,1176,84,076B,45,7995,8000,34C07B,104B6D,0E1557,
002A38,099,040*35
$GLMLA,16,09,78,1176,84,066C,46,78CF,8000,34C07B,0663F0,1A6239,
0029DF,099,030*38
$GLMLA,16,10,79,1176,80,0AFC,45,8506,8000,34C057,08DE48,1C44CA,
0029D7,099,000*6B
$GLMLA,16,11,82,1176,8A,12D3,0F,E75D,8000,34BE85,10AEA6,1781B7,
00235A,099,207*6E
$GLMLA,16,12,83,1176,03,0866,0F,6C08,8000,34C009,11F32E,18839D,
002B22,099,214*36
$GLMLA,16,13,85,1176,88,01A6,0D,9DC9,8000,34BFF8,031887,02DA1E,
002838,099,242*6D
$GLMLA,16,14,86,1176,8A,00E1,0E,4B15,8000,34C016,058181,010433,
0027F0,099,227*6F
$GLMLA,16,15,87,1176,03,0383,0F,824C,8000,34BFDA,081864,1104EA,
002B04,099,00C*60
```

```
$GLMLA,16,16,88,1176,02,0821,0F,8AC8,8000,34C05B,0A8510,12DCB6,
002B6F,099,020*3F
```
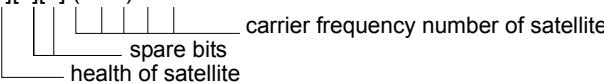
> ⓘ Refer to the GLONASS section of our book [An Introduction to GNSS](), available on our website at [www.novatel.com/support/]().

Field	Structure	Field Description	Symbol	Example
1	$GLMLA	Log header		$GLMLA
2	#alm	Number of NMEA almanac messages in the set	x.x	16
3	alm#	Current message number	x.x	13
4	slot	Slot number for satellite (65-96) [a]	xx	85
5	N	Calendar day count within the four year period from the last leap year	x.x	1176
6	hlth & freq	Health and frequency for satellite [b]	hh	88
7	ecc	Eccentricity [c]	hhhh	01a6
8	ΔTdot	Rate of change of orbital period (s/orbital period^2) [c]	hh	0d
9	w	Argument of perigee (PZ-90.02), in radians [c]	hhhh	9dc9
10	t_{16MSB}	Clock offset, in seconds [c]	hhhh	8000
11	ΔT	Correction to the mean value of the Draconian period (s/orbital period) [c]	hhhhhh	34bff8
12	tλ	GLONASS Time of ascending node equator crossing, in seconds [c]	hhhhhh	031887
13	l	Longitude of ascending node equator crossing (PZ-90.02), in radians [c]	hhhhhh	02da1e
14	Δi	Correction to nominal inclination, in radians [c]	hhhhhh	002838
15	t_{12LSB}	Clock offset, in seconds [c]	hhh	099
16	t	Coarse value of the time scale shift [c]	hhh	242
17	xxxx	32-bit CRC (ASCII and Binary only)	Hex	*6D
18	[CR][LF]	Sentence terminator (ASCII only)	-	[CR][LF]

a. The NMEA GLONASS PRN numbers are 64 plus the GLONASS slot number. Current slot numbers are 1 to 24 which give the range 65 to 88. PRN numbers 89 to 96 are available if slot numbers above 24 are allocated to on-orbit spares.

b. Health and carrier frequency numbers are represented in this 2-character Hex field as:

hh = [8][7][6][5][4][3][2][1] (LSB)

   carrier frequency number of satellite
   spare bits
   health of satellite

c. The LSB of the Hex data field corresponds to the LSB of the word indicated in the Table 4.3 of the GLONASS Interface Control Document, 1995. If the number of available bits in the Hex field is greater than the word, the MSB (upper bits) are unused and filled with zeroes.

3.2.36 GLOALMANAC
Decoded GLONASS Almanac

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The GLONASS almanac reference time and week are in GPS reference time coordinates. GLONASS ephemeris information is available through the GLOEPHEMERIS log (*page 461*).

Nominal orbit parameters of the GLONASS satellites are as follows:

- Draconian period - 11 hours 15 minutes 44 seconds (see fields 14 and 15 in the following table)
- Orbit altitude - 19100 km
- Inclination - 64.8 (see field 11)
- Eccentricity - 0 (see field 12)

Message ID: **718**
Log Type: **Asynch**

Recommended Input:

log gloalmanaca onchanged

ASCII Example:

```
#GLOALMANACA,COM1,0,52.5,SATTIME,1364,410744.000,00000000,BA83,
2310;
24,
1364,336832.625,1,2,0,0,2018.625000000,-2.775537500,0.028834045,
0.001000404,2.355427500,-2656.076171875,0.000000000,0.000091553,
1364,341828.437,2,1,0,0,7014.437500000,-3.122226146,0.030814438,
0.004598618,1.650371580,-2656.160156250,0.000061035,0.000095367,
1364,347002.500,3,12,0,0,12188.500000000,2.747629236,0.025376596,
0.002099991,-2.659059822,-2656.076171875,-0.000061035,-0.000198364,
1364,351887.125,4,6,0,0,17073.125000000,2.427596502,0.030895332,
0.004215240,1.438586358,-2656.167968750,-0.000061035,0.000007629,
.
.
.
1364,364031.187,23,11,0,1,29217.187500000,0.564055522,0.030242192,
0.001178741,2.505278248,-2655.957031250,0.000366211,0.000019073,
1364,334814.000,24,3,0,1,0.000000000,0.000000000,0.000000000,0.0000
00000,0.000000000,0.000000000,0.000000000,0.000000000*4DC981C7
```

> Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLOALMANAC header	Log header		H	0
2	#recs	The number of GLONASS almanac records to follow. Set to zero until almanac data is available	Long	4	H
3	week	GPS reference week, in weeks	Ulong	4	H+4
4	time	GPS reference time, in milliseconds (binary data) or seconds (ASCII data)	GPSec	4	H+8
5	slot	Slot number for satellite, ordinal	Uchar	1	H+12
6	frequency	Frequency for satellite, ordinal (frequency channels are in the range -7 to +6)	Char	1	H+13
7	sat type	Satellite type where 　0 = GLO_SAT 　1 = GLO_SAT_M (M type) 　　2 = GLO_SAT_K (K type)	Uchar	1	H+14
8	health	Almanac health where 　0 = GOOD 　1 = BAD	Uchar	1	H+15
9	TlambdaN	GLONASS Time of ascending node equator crossing, in seconds	Double	8	H+16
10	lambdaN	Longitude of ascending node equator crossing (PZ-90.02), in radians	Double	8	H+24
11	deltaI	Correction to nominal inclination, in radians	Double	8	H+32
12	ecc	Eccentricity	Double	8	H+40
13	ArgPerig	Argument of perigee (PZ-90.02), in radians	Double	8	H+48
14	deltaT	Correction to the mean value of the Draconian period (s/orbital period)	Double	8	H+56
15	deltaTD	Rate of change of orbital period (s/orbital period^2)	Double	8	H+64
16	tau	Clock offset, in seconds	Double	8	H+72
17...	Next message offset = H + 4 + (#recs x 76)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H + 4 + (76 x #recs)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.37 GLOCLOCK
GLONASS clock information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the time difference information between GPS and GLONASS time as well as status flags. The status flags are used to indicate the type of time processing used in the least squares adjustment. GPS and GLONASS time are both based on the Universal Time Coordinated (UTC) time scale with some adjustments. GPS reference time is continuous and does not include any of the leap second adjustments to UTC applied since 1980. The result is that GPS reference time currently leads UTC time by 15 seconds.

GLONASS time applies leap seconds but is also three hours ahead to represent Moscow time. The nominal offset between GPS and GLONASS time is therefore due to the three hour offset minus the leap second offset. As well as the nominal offset, there is a residual offset on the order of nanoseconds which must be estimated in the least squares adjustment. The GLONASS-M satellites broadcasts this difference in the navigation message.

This log also contains information from the GLONASS navigation data relating GLONASS time to UTC.

Message ID: 719
Log Type: Asynch

Recommended Input:

```
log gloclocka onchanged
```

ASCII Example:

```
#GLOCLOCKA,COM1,0,54.5,SATTIME,1364,411884.000,00000000,1D44,2310;
0,0.000000000,0.000000000,0,0,-0.000000275,792,-0.000001207,
0.000000000,0.000000000,0*437E9AFAF
```

> (i) Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLOCLOCK header	Log header		H	0
2			Ulong	4	H
3	Reserved		Double	8	H+4
4			Double	8	H+12
5	sat type	Satellite type where 0 = GLO_SAT 1 = GLO_SAT_M (M type) 2 = GLO_SAT_K (K type)	Uchar	1	H+20

6	N^4	Four-year interval number starting from 1996	Uchar	1 [a]	H+21 [a]
7	τ_{GPS}	Correction to GPS time relative to GLONASS time	Double	8	H+24
8	N^A	GLONASS calendar day number within a four year period beginning since the leap year, in days	Ushort	2 [a]	H+32 [a]
9	τ_C	GLONASS time scale correction to UTC(SU) given at beginning of day N4, in seconds	Double	8	H+36
10	b1	Beta parameter 1st order term	Double	8	H+44
11	b2	Beta parameter 2nd order term	Double	8	H+52
12	Kp	Kp provides notification of the next expected leap second. For more information, see *Table 102*	Uchar	1	H+60
13	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+61
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

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

Table 102: Kp UTC Leap Second Descriptions

Kp	Information on UTC Leap Second[a]
00	No UTC update for this quarter
01	UTC update of plus 1 second at the end of current quarter
11	UTC update of minus 1 second at end of current quarter

a. Based on GLONASS ICD version 5.1, 2008.

3.2.38 GLOEPHEMERIS
Decoded GLONASS ephemeris

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains GLONASS ephemeris information. GLONASS ephemerides are referenced to the PZ90.02 geodetic datum. No adjustment between the GPS and GLONASS reference frames are made for positioning. Multiple messages are transmitted, one for each SVID with data.

Message ID:	**723**
Log Type:	**Asynch**

Recommended Input:

    log gloephemerisa onchanged

Example:

```
#GLOEPHEMERISA,COM1,3,49.0,SATTIME,1364,413624.000,00000000,6B64,23
10;43,8,1,0,1364,413114000,10786,792,0,0,87,0,9.0260864257812500E+0
6,-6.1145468750000000E+06,2.2926090820312500E+07,1.4208841323852539
E+03,2.8421249389648438E+03,1.9398689270019531E+02,0.00000000000000
000,-2.79396772384643555E-06,-2.79396772384643555E-06,2.12404876947
402954E-04,-1.396983862E-08,-3.63797880709171295E-12,78810,3,15,0,
12*A02CE18B
```

```
#GLOEPHEMERISA,COM1,2,49.0,SAT-
TIME,1364,413626.000,00000000,6B64,2310;44,11,1,0,1364,413116000,10
784,792,0,0,87,13,-1.2882617187500000E+06,-1.9318657714843750E+07,
1.6598909179687500E+07,9.5813846588134766E+02,2.0675134658813477E+0
3,2.4769935607910156E+03,2.79396772384643555E-06,-3.725290298461914
06E-06,-1.86264514923095703E-06,6.48368149995803833E-05,-4.6566128
73E-09,3.63797880709171295E-12,78810,3,15,3,28*E2D5EF15
```

```
#GLOEPHEMERISA,COM1,1,49.0,SATTIME,1364,413624.000,00000000,6B64,23
10;45,13,0,0,1364,413114000,10786,0,0,0,87,0,-1.1672664062500000E+
07,-2.2678505371093750E+07,4.8702343750000000E+05,-1.17333412170410
16E+02,1.3844585418701172E+02,3.5714883804321289E+03,2.793967723846
43555E-06,-2.79396772384643555E-06,0.00000000000000000,-4.53162938
356399536E-05,5.587935448E-09,-2.36468622460961342E-11,78810,0,0,
0,8*C15ABFEB
```

```
#GLOEPHEMERISA,COM1,0,49.0,SATTIME,1364,413624.000,00000000,6B64,
2310;59,17,0,0,1364,413114000,10786,0,0,0,87,0,-2.3824853515625000
E+05,-1.6590188964843750E+07,1.9363733398437500E+07,1.35170745849
60938E+03,-2.2859592437744141E+03,-1.9414072036743164E+03,1.86264
514923095703E-06,-3.72529029846191406E-06,-1.86264514923095703E-06,
7.92574137449264526E-05,4.656612873E-09,2.72848410531878471E-12,
78810,0,0,0,12*ED7675F5
```

> ⓘ Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

Table 103: GLONASS Ephemeris Flags Coding

N 0	<- Nibble Number			
4 3 2 1 0 Bit	Description	Range Values	Hex Value	
lsb = 0	P1 FLAG - TIME INTERVAL BETWEEN ADJACENT iISSUE(tb) VALUES	See Table below (*Table 104*)	00000001	
1			00000002	
2	P2 FLAG - ODDNESS OR EVENNESS OF iISSUE(tb) VALUE	0 = even, 1 = odd	00000004	
3	P3 FLAG - NUMBER OF SATELLITES WITH ALMANAC INFORMATION WITHIN CURRENT SUBFRAME	0 = four, 1 = five	00000008	
4 : 31	RESERVED (N-1 through N-7)			

Table 104: *Bits 0 - 1: P1 Flag Range Values*

State	Description
00	0 minutes
01	30 minutes
10	45 minutes
11	60 minutes

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLO-EPHEMERIS header	Log header		H	0
2	sloto	Slot information offset - PRN identification (Slot + 37). This is also called SLOTO in Connect	Ushort	2	H
3	freqo	Frequency channel offset for satellite in the range 0 to 20	Ushort	2	H+2
4	sat type	Satellite type where 0 = GLO_SAT 1 = GLO_SAT_M (M type) 2 = GLO_SAT_K (K type)	Uchar	1	H+4
5	Reserved			1	H+5
6	e week	Reference week of ephemeris (GPS reference time)	Ushort	2	H+6
7	e time	Reference time of ephemeris (GPS reference time) in ms	Ulong	4	H+8
8	t offset	Integer seconds between GPS and GLONASS time. A positive value implies GLONASS is ahead of GPS reference time.	Ulong	4	H+12
9	Nt	Calendar number of day within 4 year interval starting at Jan 1 of a leap year	Ushort	2	H+16
10	Reserved			1	H+18
11				1	H+19
12	issue	15 minute interval number corresponding to ephemeris reference time	Ulong	4	H+20
13	health[a]	Ephemeris health where 0-3 = GOOD 4-15 = BAD	Ulong	4	H+24
14	pos x	X coordinate for satellite at reference time (PZ-90.02), in metres	Double	8	H+28
15	pos y	Y coordinate for satellite at reference time (PZ-90.02), in metres	Double	8	H+36
16	pos z	Z coordinate for satellite at reference time (PZ-90.02), in metres	Double	8	H+44
17	vel x	X coordinate for satellite velocity at reference time (PZ-90.02), in metres/s	Double	8	H+52
18	vel y	Y coordinate for satellite velocity at reference time (PZ-90.02), in metres/s	Double	8	H+60
19	vel z	Z coordinate for satellite velocity at reference time (PZ-90.02), in metres/s	Double	8	H+68
20	LS acc x	X coordinate for lunisolar acceleration at reference time (PZ-90.02), in metres/s/s	Double	8	H+76

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
21	LS acc y	Y coordinate for lunisolar acceleration at reference time (PZ-90.02), in metres/s/s	Double	8	H+84
22	LS acc z	Z coordinate for lunisolar acceleration at reference time (PZ-90.02), in metres/s/s	Double	8	H+92
23	tau_n	Correction to the nth satellite time t_n relative to GLONASS time t_c, in seconds	Double	8	H+100
24	delta_tau_n	Time difference between navigation RF signal transmitted in L2 sub-band and navigation RF signal transmitted in L1 sub-band by nth satellite, in seconds	Double	8	H+108
25	gamma	Frequency correction, in seconds/second	Double	8	H+116
26	Tk	Time of frame start (since start of GLONASS day), in seconds	Ulong	4	H+124
27	P	Technological parameter	Ulong	4	H+128
28	Ft	User range	Ulong	4	H+132
29	age	Age of data, in days	Ulong	4	H+136
30	Flags	Information flags, see *Table 103, GLONASS Ephemeris Flags Coding* on *page 462*	Ulong	4	H+140
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+144
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. The last four bits of this field are used to describe the health.
    Bit 0-2: Bn
    Bit 3: In
    All other bits are reserved and set to 0.

3.2.39 GLORAWALM
Raw GLONASS Almanac data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw almanac subframes as received from the GLONASS satellite.

Message ID: **720**
Log Type: **Asynch**

Recommended Input:

```
log glorawalma onchanged
```

Example:

```
#GLORAWALMA,COM1,0,44.5,SATTIME,1364,419924.000,00000000,77BB,2310;
1364,419954.069,54,
0563100000A4000000006F,0,
0681063C457A12CC0419BE,0,
075FF807E2A69804E0040B,0,
0882067FCD80141692D6F2,0,
09433E1B6676980A40429B,0,
0A838D1BFCB4108B089A8C,0,
0BEC572F9C869804F05882,0,
.
.
.
06950201E02E13D3819564,0,
07939A4A16FE97FE814AD0,0,
08960561CECC13B0014613,0,
09469A5D70C69802819466,0,
0A170165BED413B704D416,0,
0B661372213697FD41965A,0,
0C18000000000000000006,0,
0D00000000000000000652,0,
0E000000000000000000D0,0*B516623B
```

> ℹ️ Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLORAWALM header	Log header		H	0
2	week	GPS reference week, in weeks	Ulong	4	H
3	time	GPS reference time, in milliseconds (binary data) or seconds (ASCII data)	GPSec	4	H+4
4	#recs	Number of records to follow	Ulong	4	H+8
5	string	GLONASS data string	String [11]	11	H+12
6	Reserved		Uchar	1	H+23
7...	Next record offset = H+8+(#recs x 12)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+12+ (#recsx12)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.40 GLORAWEPHEM
Raw GLONASS Ephemeris data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw ephemeris frame data as received from the GLONASS satellite.

Message ID: 792
Log Type: Asynch

Recommended Input:

> log glorawephema onchanged

Example:

```
#GLORAWEPHEMA,COM1,3,47.0,SATTIME,1340,398653.000,00000000,332D,
2020;38,9,0,1340,398653.080,4,0148D88460FC115DBDAF78,0,0218E0033667
AEC83AF2A5,0,038000B9031E14439C75EE,0,0404F22660000000000065,0*17F3
DD17
...
#GLORAWEPHEMA,COM1,0,47.0,SATTIME,1340,398653.000,00000000,332D,
2020;41,13,0,1340,398653.078,4,0108D812532805BFA1CD2C,0,0208E0A36E8
E0952B111-
DA,0,03C02023B68C9A32410958,0,0401FDA44000000000002A,0*0B237405
```

> ⓘ Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLORAWEPHEM header	Log header		H	0
2	sloto	Slot information offset - PRN identification (Slot + 37). Ephemeris relates to this slot and is also called SLOTO in NovAtel Connect	Ushort	2	H
3	freqo	Frequency channel offset in the range 0 to 20	Ushort	2	H+2
4	sigchan	Signal channel number	Ulong	4	H+4
5	week	GPS reference week, in weeks	Ulong	4	H+8
6	time	GPS reference time, in milliseconds (binary data) or seconds (ASCII data)	GPSec	4	H+12
7	#recs	Number of records to follow	Ulong	4	H+16
8	string	GLONASS data string	String [11]	11	H+20
9	Reserved		Uchar	1	H+31
10...	Next record offset = H+20+(#recs x 12)				

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
variable	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+20+ (#recsx12)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.41 GLORAWFRAME
Raw GLONASS frame data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw GLONASS frame data as received from the GLONASS satellite. Multiple messages are transmitted, one for each SVID with data.

Message ID: **721**
Log Type: **Asynch**

Recommended Input:

> log glorawframea onchanged

Example:
```
#GLORAWFRAMEA,COM1,19,53.0,SATTIME,1340,398773.000,00000000,8792,
2020;3,39,8,1340,398773.067,44,44,15,0148DC0B67E9184664CB35,0,
0218E09DC8A3AE8C6BA18D,0,
...
0F00000000000000000000,0*11169F9E
...
#GLORAWFRAMEA,COM1,0,53.0,SATTIME,1340,398713.000,00000000,8792,
2020;1,41,13,1340,398713.077,36,36,15,0108DA12532805BFA1CDED,0,
0208E0A36E8E0952B111DA,0,03C02023B68C9A32410958,0,
...
0F6EFB59474697FD72C4E2,0*0A6267C8
```

> ℹ Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLORAWFRAME header	Log header		H	0
2	frame#	Frame number	Ulong	4	H
3	sloto	Slot information offset - PRN identification (Slot + 37). Ephemeris relates to this slot and is also called SLOTO in CDU.	Ushort	2	H+4
4	freqo	Frequency channel offset in the range 0 to 20	Ushort	2	H+6
5	week	GPS Week, in weeks	Ulong	4	H+8
6	time	GPS Time, in milliseconds (binary data) or seconds (ASCII data)	GPSec	4	H+12
7	frame decode	Frame decoder number	Ulong	4	H+16

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
8	sigchan	Signal channel number	Ulong	4	H+20
9	#recs	Number of records to follow	Ulong	4	H+24
10	string	GLONASS data string	String [11]	11	H+28
11	Reserved		Uchar	1	H+39
12...	Next record offset = H+28+ (#recs x 12)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H +28+ (#recs x 1)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.42 *GLORAWSTRING*
Raw GLONASS string

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw string data as received from the GLONASS satellite.

Message ID: **722**
Log Type: **Asynch**

Recommended Input:

```
log glorawstringa onchanged
```

Example:

```
#GLORAWSTRINGA,COM1,0,51.0,SATTIME,1340,399113.000,00000000,50AC,
2020;4,6,061000000000000000004F,0*5B215FB2
```

> ⓘ   Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLORAWSTRING header	Log header		H	0
2	slot	Slot identification	Uchar	1	H
3	freq	Frequency channel (frequency channels are in the range -7 to +13)	Char	1	H+2
4	string	GLONASS data string	Uchar [11]	11	H+4
5	Reserved		Uchar	1	H+15
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+16
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.43 GPALM
Almanac data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log outputs raw almanac data for each GPS satellite PRN contained in the broadcast message. A separate record is logged for each PRN, up to a maximum of 32 records. GPALM outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. UTC time is then set to VALID. It takes a minimum of 12.5 minutes to collect a complete almanac following receiver boot-up. In the case of a GLONASS capable receiver, the UTC offset can be determined once the GLONASS ephemeris is decoded, which takes about 50 seconds. If an almanac was stored in NVM, the stored values are reported in the GPALM log once time is set on the receiver.

> To obtain copies of ICD-GPS-200, refer to ARINC on our website at www.novatel.com/support/knowledge-and-learning/published-papers-and-documents/standards-and-references/. NMEA contact information is also located there.

Message ID: **217**
Log Type: **Asynch**

Recommended Input:

> log gpalm onchanged

Example:
```
$GPALM,28,01,01,1337,00,305A,90,1B9D,FD5B,A10CE9,BA0A5E,2F48F1,CCCB
76,006,001*27
$GPALM,28,02,02,1337,00,4AA6,90,0720,FD50,A10C5A,4DC146,D89BAB,0790
B6,FE4,000*70
 .
 .
 .
$GPALM,28,24,26,1337,00,878C,90,1D32,FD5C,A10C90,1DB6B6,2EB7F5,CE95
C8,00D,000*23
$GPALM,28,25,27,1337,00,9CDE,90,07F2,FD54,A10-
DA5,ADC097,562DA3,6488DD,00E,000*2F
$GPALM,28,26,28,1337,00,5509,90,0B7C,FD59,A10CC4,A1D262,83E2C0,3003
BD,02D,000*78
$GPALM,28,27,29,1337,00,47F7,90,1B20,FD58,A10CE0,D40A0B,2D570E,2216
41,122,006*7D
$GPALM,28,28,30,1337,00,4490,90,0112,FD4A,A10CC1,33D10A,81DFC5,3BDB
0F,178,004*28
```

> See the GPGGA usage box that applies to all NMEA logs on .

Field	Structure	Field Description	Symbol	Example
1	$GPALM	Log header		$GPALM
2	# msg	Total number of messages logged. Set to zero until almanac data is available	x.x	17
3	msg #	Current message number	x.x	17
4	PRN	Satellite PRN number: 　　GPS　　= 1 to 32	xx	28
5	GPS wk	GPS reference week number [a]	x.x	653
6	SV hlth	SV health, bits 17-24 of each almanac page [b]	hh	00
7	ecc	e, eccentricity [c][d]	hhhh	3EAF
8	alm ref time	to a almanac reference time [c]	hh	87
9	incl angle	$(\text{sigma})_i$, inclination angle [c]	hhhh	OD68
10	omegadot	OMEGADOT, rate of right ascension [c]	hhhh	FD30
11	rt axis	$(A)^{1/2}$, root of semi-major axis [c]	hhhhhh	A10CAB
12	omega	omega, argument of perigee [c][e]	hhhhhh	6EE732
13	long asc node	$(\text{OMEGA})_o$, longitude of ascension node [c]	hhhhhh	525880
14	M_O	Mo, mean anomaly [c]	hhhhhh	6DC5A8
15	a_{f0}	af0, clock parameter [c]	hhh	009
16	a_{f1}	af1, clock parameter [c]	hhh	005
17	*xx	Checksum	*hh	*37
18	[CR][LF]	Sentence terminator		[CR][LF]

a　Variable length integer, 4-digits maximum from (2) most significant binary bits of Subframe 1, Word 3 reference Table 20-I, ICD-GPS-200, Rev. B, and (8) least significant bits from subframe 5, page 25, word 3 reference Table 20-I, ICD-GPS-200.

b　Reference paragraph 20.3.3.5.1.3, Table 20-VII and Table 20-VIII, ICD-GPS-200, Rev. B.

c　Reference Table 20-VI, ICD-GPS-200, Rev. B for scaling factors and units.

d　A quantity defined for a conic section where e= 0 is a circle, e = 1 is an ellipse, 0<e<1 is a parabola and e>1 is a hyperbola.

e　A measurement along the orbital path from the ascending node to the point where the SV is closest to the Earth, in the direction of the SV's motion.

3.2.44   GPGGA
GPS fix data and undulation

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains time, position and fix related data of the GNSS receiver. For greater precision but with loss of the undulation fields, use the GPGGARTK log (*page 479*). See also *Table 106, Position Precision of NMEA Logs* on *page 481*.

The GPGGA log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

Message ID: **218**
Log Type   **Synch**

Recommended Input:

         log gpgga ontime 1

Example:
```
$GPGGA,134658.00,5106.9792,N,11402.3003,W,2,09,1.0,1048.47,M,
-16.27,M,08,AAAA*60
```

> (i) The NMEA (National Marine Electronics Association) has defined standards that specify how electronic equipment for marine users communicate. GNSS receivers are part of this standard and the NMEA has defined the format for several GNSS data logs otherwise known as 'sentences'.
>
> Each NMEA sentence begins with a '$' followed by the prefix 'GL' or 'GN'followed by a sequence of letters that define the type of information contained in the sentence. Data contained within the sentence is separated by commas and the sentence is terminated with a two digit checksum followed by a carriage return/line feed. Here is an example of a NMEA sentence describing time, position and fix related data:
>
> $GPGGA,134658.00,5106.9792,N,11402.3003,W,2,09,1.0,1048.47,M,
>
> -16.27,M,08,AAAA*60
>
> The GPGGA sentence shown above and other NMEA logs are output the same no matter what GNSS receiver is used, providing a standard way to communicate and process GNSS information. For more information about NMEA, see the NMEATALKER command on page *219.*

Field	Structure	Field Description	Symbol	Example
1	$GPGGA	Log header		$GPGGA
2	utc	UTC time status of position (hours/minutes/ seconds/ decimal seconds)	hhmmss.ss	202134.00
3	lat	Latitude (DDmm.mm)	llll.ll	5106.9847
4	lat dir	Latitude direction (N = North, S = South)	a	N
5	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.2986
6	lon dir	Longitude direction (E = East, W = West)	a	W
7	quality	refer to *Table 105, GPS Quality Indicators* on *page 476*	x	1
8	# sats	Number of satellites in use. May be different to the number in view	xx	10
9	hdop	Horizontal dilution of precision	x.x	1.0
10	alt	Antenna altitude above/below mean sea level	x.x	1062.22
11	a-units	Units of antenna altitude (M = metres)	M	M
12	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid	x.x	-16.271
13	u-units	Units of undulation (M = metres)	M	M
14	age	Age of correction data (in seconds) [a]	xx	(empty when no differential data is present)
15	stn ID	Differential base station ID	xxxx	(empty when no differential data is present)
16	*xx	Checksum	*hh	*48
17	[CR][LF]	Sentence terminator		[CR][LF]

a. The maximum age reported here is limited to 99 seconds.

Table 105: GPS Quality Indicators

Indicator	Field Description
0	Fix not available or invalid
1	Single point
2	Pseudorange differential
	Unconverged OmniSTAR
	HP/XP/G2/VBS converging PPP
4	RTK fixed ambiguity solution (RT2)
	Operational (refer to *UALCONTROL* on *page 331*)
5	RTK floating ambiguity solution (RT20)
	Converged OmniSTAR HP/XP/G2
	Converged PPP
6	Dead reckoning mode
7	Manual input mode (fixed position)
8	Simulator mode
9	WAAS (SBAS)[a]

a. An indicator of 9 has been temporarily set for SBAS (NMEA standard for SBAS not decided yet). This indicator can be customized using the GGAQUALITY command.

Refer to the BESTPOS command (*page 406*) and *Table 89, Supplemental Position Types and NMEA Equivalents* on *page 411*.

3.2.45 GPGGALONG
Fix data, extra precision and undulation

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains, time, position, undulation and fix related data of the GNSS receiver. This is output as a GPGGA log but the GPGGALONG log differs from the normal GPGGA log by its extra precision. See also *Table 106, Position Precision of NMEA Logs* on *page 481.*

The GPGGALONG log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

Message ID: **521**
Log Type: **Synch**

Recommended Input:

> log gpggalong ontime 1

Example 1:

```
$GPGGA,181126.00,5106.9802863,N,11402.3037304,W,7,11,0.9,1048.234,
M,-16.27,M,,*51
```

Example 2:

```
$GPGGA,134658.00,5106.9802863,N,11402.3037304,W,2,09,1.0,1048.234,
M,-16.27,M,08,AAAA
```

> ⓘ   See the GPGGA usage box that applies to all NMEA logs on *page 474*.

Field	Structure	Field Description	Symbol	Example
1	$GPGGALONG	Log header		$GPGGA
2	utc	UTC time status of position (hours/minutes/ seconds/ decimal seconds)	hhmmss.ss	202126.00
3	lat	Latitude (DDmm.mm)	llll.ll	5106.9847029
4	lat dir	Latitude direction (N = North, S = South)	a	N
5	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.2986286
6	lon dir	Longitude direction (E = East, W = West)	a	W
7	GPS qual	Refer to *Table 105, GPS Quality Indicators* on *page 476*	x	1
8	# sats	Number of satellites in use (00-12). May be different to the number in view	xx	10
9	hdop	Horizontal dilution of precision	x.x	1.0
10	alt	Antenna altitude above/below msl	x.x	1062.376

Field	Structure	Field Description	Symbol	Example
11	units	Units of antenna altitude (M = metres)	M	M
12	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid	x.x	-16.271
13	u-units	Units of undulation (M = metres)	M	M
14	age	Age of Differential GPS data (in seconds) [a]	xx	10 *(empty when no differential data is present)*
15	stn ID	Differential base station ID, 0000-1023	xxxx	AAAA (empty when no differential data is present)
16	*xx	Checksum	*hh	*48
17	[CR][LF]	Sentence terminator		[CR][LF]

a. The maximum age reported here is limited to 99 seconds.

> (i) Refer to the BESTPOS command (*page 406*) and *Table 89, Supplemental Position Types and NMEA Equivalents* on *page 411*.

3.2.46 GPGGARTK
Global position system fix data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains time, position and fix-related data of the GNSS receiver. This is output as a GPGGA log but the GPGGARTK log differs from the normal GPGGA log by its extra precision. In order for the position to be output with this extra precision, the undulation fields are unavailable (see the GPGGA log on *page 474*). See also *Table 106, Position Precision of NMEA Logs* on *page 481.*

The GPGGARTK log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

Message ID: **259**
Log Type: **Synch**

Recommended Input:

> log gpggartk ontime 1

Example:
```
$GPG-
GA,135324.00,5106.9791988,N,11402.3002127,W,2,09,1.0,1047.606,M,,,
04,AAAA*1C
```

> (i) The GPGGARTK log is ideal for RTK positioning applications where millimeter level position precision is required. See also the GPGGA usage box that applies to all NMEA logs on *page 474*.

Field	Structure	Field Description	Symbol	Example
1	$GPGGA	Log header		$GPGGA
2	utc	UTC time status of position (hours/minutes/ seconds/ decimal seconds)	hhmmss.ss	220147.50
3	lat	Latitude (DDmm.mm)	llll.ll	5106.7194489
4	lat dir	Latitude direction (N = North, S = South)	a	N
5	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.3589020
6	lon dir	Longitude direction (E = East, W = West)	a	W
7	GPS qual	Refer to *Table 105, GPS Quality Indicators* on *page 476*	x	1
8	# sats	Number of satellites in use. May be different to the number in view	xx	08
9	hdop	Horizontal dilution of precision	x.x	0.9
10	alt	Antenna altitude above/below mean sea level	x.x	1080.406

Field	Structure	Field Description	Symbol	Example
11	units	Units of antenna altitude (M = metres)	M	M
12	null	(This field not available on OEM6 family receivers)		(empty when no differential data is present)
13	null	(This field not available on OEM6 family receivers)		
14	age	Age of Differential GPS data (in seconds) [a]	xx	
15	stn ID	Differential base station ID, 0000-1023	xxxx	
16	*xx	Checksum	*hh	*48
17	[CR][LF]	Sentence terminator		[CR][LF]

a. The maximum age reported here is limited to 99 seconds.

ⓘ Refer to the BESTPOS command (*page 406*) and *Table 89, Supplemental Position Types and NMEA Equivalents* on *page 411*.

3.2.47 GPGLL
Geographic position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains latitude and longitude of present vessel position, time of position fix and status.

Table 106 compares the position precision of selected NMEA logs.

The GPGLL log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

> (i) If the `NMEATALKER` command, see *page 219*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only) or GN (satellites from both systems) or GA (Galileo satellites only).

Message ID: **219**
Log Type: **Synch**

Recommended Input:

    log gpgll ontime 1

Example 1 (GPS only):

    $GPGLL,5107.0013414,N,11402.3279144,W,205412.00,A,A*73

Example 2 (Combined GPS and GLONASS):

    $GNGLL,5107.0014143,N,11402.3278489,W,205122.00,A,A*6E

Table 106: Position Precision of NMEA Logs

NMEA Log	Latitude (# of decimal places)	Longitude (# of decimal places)	Altitude (# of decimal places)
GPGGA	4	4	2
GPGGALONG	7	7	3
GPGGARTK	7	7	3
GPGLL	7	7	N/A
GPRMC	7	7	N/A

> (i) See the GPGGA usage box that applies to all NMEA logs on *page 474*.

Field	Structure	Field Description	Example
1	$GPGLL	Log header	$GPGLL
2	lat	Latitude (DDmm.mm)	5106.7198674
3	lat dir	Latitude direction (N = North, S = South)	N
4	lon	Longitude (DDDmm.mm)	11402.3587526
5	lon dir	Longitude direction (E = East, W = West)	W
6	utc	UTC time status of position (hours/minutes/seconds/ decimal seconds)	220152.50
7	data status	Data status: A = Data valid, V = Data invalid	A
8	mode ind	Positioning system mode indicator, see *Table 107* on *page 492*	A
9	*xx	Checksum	*1B
10	[CR][LF]	Sentence terminator	[CR][LF]

3.2.48   GPGRS
GPS range residuals for each satellite

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Range residuals can be computed in two ways, and this log reports those residuals. Under mode 0, residuals output in this log are used to update the position solution output in the GPGGA message. Under mode 1, the residuals are recomputed after the position solution in the GPGGA message is computed. The receiver computes range residuals in mode 1. An integrity process using GPGRS would also require GPGGA (for position fix data), GPGSA (for DOP figures) and GPGSV (for PRN numbers) for comparative purposes.

The GPGRS log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

> (i)
> 1. If the range residual exceeds ± 99.9, then the decimal part is dropped. Maximum value for this field is ± 999. The sign of the range residual is determined by the order of parameters used in the calculation as follows:
>   - range residual = calculated range - measured range
> 2. If the NMEATALKER command, see *page 219*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only) or GN (satellites from both systems) or GA (Galileo satellites only).
> 3. There is no residual information available from the OmniSTAR HP/XP/G2 service, so the GPGRS contains the pseudorange position values when using it. For the OmniSTAR VBS service, residual information is available.

Message ID: **220**
Log Type: **Synch**

Recommended Input:

> log gpgrs ontime 1

Example 1 (GPS only):

```
$GPGRS,142406.00,1,-1.1,-0.1,1.7,1.2,-2.0,-0.5,1.2,-1.2,-0.1,,,*67
```

Example 2 (Combined GPS and GLONASS):

```
$GNGRS,143209.00,1,-0.2,-0.5,2.2,1.3,-2.0,-1.3,1.3,-0.4,-1.2,-0.2
,,*72
$GNGRS,143209.00,1,1.3,-6.7,,,,,,,,,,*73
```

> (i)   See the GPGGA usage box that applies to all NMEA logs on *page 474.*

Field	Structure	Field Description	Symbol	Example
1	$GPGRS	Log header		$GPGRS
2	utc	UTC time status of position (hours/ minutes/seconds/decimal seconds)	hhmmss.ss	192911.0
3	mode	Mode 0= residuals were used to calculate the position given in the matching GGA line (apriori) (not used by OEM6 receivers) Mode 1= residuals were recomputed after the GGA position was computed (preferred mode)	x	1
4 - 15	res	Range residuals for satellites used in the navigation solution. Order matches order of PRN numbers in GPGSA	x.x,x.x,.....	-13.8,-1.9,11.4,-33.6,0.9, 6.9,-12.6,0.3,0.6, -22.3
16	*xx	Checksum	*hh	*65
17	[CR][LF]	Sentence terminator		[CR][LF]

3.2.49 GPGSA
GPS DOP and active satellites

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains GNSS receiver operating mode, satellites used for navigation and DOP values.

The GPGSA log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

> ⓘ
> 1. If the DOP values exceed 9999.0 or there is an insufficient number of satellites to calculate a DOP value, 9999.0 is reported for PDOP and HDOP. VDOP is reported as 0.0 in this case.
> 2. If the NMEATALKER command, see *page 219*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), or GN (satellites from both systems) or GA (Galileo satellites only).

Message ID: **221**
Log Type: **Synch**

Recommended Input:

log gpgsa ontime 1

Example 1 (GPS only):

`$GPGSA,M,3,17,02,30,04,05,10,09,06,31,12,,,1.2,0.8,0.9*35`

Example 2 (Combined GPS and GLONASS):

`$GNGSA,M,3,17,02,30,04,05,10,09,06,31,12,,,1.2,0.8,0.9*2B`
`$GNGSA,M,3,87,70,,,,,,,,,,,1.2,0.8,0.9*2A`

> ⓘ
> The DOPs provide a simple characterization of the user satellite geometry. DOP is related to the volume formed by the intersection points of the user satellite vectors, with the unit sphere centered on the user. Larger volumes give smaller DOPs. Lower DOP values generally represent better position accuracy. The role of DOP in GNSS positioning is often misunderstood. A lower DOP value does not automatically mean a low position error. The quality of a GNSS derived position estimate depends upon both the measurement geometry as represented by DOP values and range errors caused by signal strength, ionospheric effects, multipath and so on.
> Also see the GPGGA usage box that applies to all NMEA logs on *page 474*.

Field	Structure	Field Description	Symbol	Example
1	$GPGSA	Log header		$GPGSA
2	mode MA	A = Automatic 2D/3D M = Manual, forced to operate in 2D or 3D	M	M
3	mode 123	Mode: 1 = Fix not available; 2 = 2D; 3 = 3D	x	3
4 - 15	prn	PRN numbers of satellites used in solution (null for unused fields), total of 12 fields GPS = 1 to 32 SBAS = 33 to 64 (add 87 for PRN number) GLO = 65 to 96 [a]	xx,xx,.....	18,03,13, 25,16, 24,12, 20,,,,
16	pdop	Position dilution of precision	x.x	1.5
17	hdop	Horizontal dilution of precision	x.x	0.9
18	vdop	Vertical dilution of precision	x.x	1.2
19	*xx	Checksum	*hh	*3F
20	[CR][LF]	Sentence terminator		[CR][LF]

a. The NMEA GLONASS PRN numbers are 64 plus the GLONASS slot number. Current slot numbers are 1 to 24 which give the range 65 to 88. PRN numbers 89 to 96 are available if slot numbers above 24 are allocated to on-orbit spares.

3.2.50 GPGST
Pseudorange measurement noise statistics

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains pseudorange measurement noise statistics are translated in the position domain in order to give statistical measures of the quality of the position solution.

This log reflects the accuracy of the solution type used in the BESTPOS (*page 406*) and GPGGA (*page 474*), except for the RMS field. The RMS field, since it specifically relates to pseudorange inputs, does not represent carrier-phase based positions. Instead it reflects the accuracy of the pseudorange position which is given in the PSRPOS log, see *page 581*.

The GPGST log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

> (i) If the NMEATALKER command, see *page 219*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only) or GN (satellites from both systems) or GA (Galileo satellites only).

Message ID: **222**
Log Type: **Synch**

Recommended Input:

       log gpgst ontime 1

Example 1 (GPS only):

   $GPGST,141451.00,1.18,0.00,0.00,0.0000,0.00,0.00,0.00*6B

Example 2 (Combined GPS and GLONASS):

   $GNGST,143333.00,7.38,1.49,1.30,68.1409,1.47,1.33,2.07*4A

ⓘ

1. See the GPGGA usage box that applies to all NMEA logs on *page 474*.

2. Accuracy is based on statistics, reliability is measured in percent. When a receiver can measure height to one metre, this is an accuracy. Usually this is a one sigma value (one SD). A one sigma value for height has a reliability of 68%, that is, the error is less than one metre 68% of the time. For a more realistic accuracy, double the one sigma value (1 m) and the result is 95% reliability (error is less than 2 m 95% of the time). Generally, GNSS heights are 1.5 times poorer than horizontal positions.

   As examples of statistics, the GPGST message and NovAtel performance specifications use Root Mean Square (RMS). Specifications may be quoted in CEP:

   - RMS - root mean square (a probability level of 68%)

   - CEP - circular error probable (the radius of a circle such that 50% of a set of events occur inside the boundary)

Field	Structure	Field Description	Symbol	Example
1	$GPGST	Log header		$GPGST
2	utc	UTC time status of position (hours/minutes/seconds/ decimal seconds)	hhmmss.ss	173653.00
3	rms	RMS value of the standard deviation of the range inputs to the navigation process. Range inputs include pseudoranges and DGPS corrections	x.x	2.73
4	smjr std	Standard deviation of semi-major axis of error ellipse (m)	x.x	2.55
5	smnr std	Standard deviation of semi-minor axis of error ellipse (m)	x.x	1.88
6	orient	Orientation of semi-major axis of error ellipse (degrees from true north)	x.x	15.2525
7	lat std	Standard deviation of latitude error (m)	x.x	2.51
8	lon std	Standard deviation of longitude error (m)	x.x	1.94
9	alt std	Standard deviation of altitude error (m)	x.x	4.30
10	*xx	Checksum	*hh	*6E
11	[CR][LF]	Sentence terminator		[CR][LF]

3.2.51 GPGSV
GPS satellites in view

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the number of SVs in view, PRN numbers, elevation, azimuth and SNR value. Four satellites maximum per message. When required, additional satellite data sent in 2 or more messages (a maximum of 9). The total number of messages being transmitted and the current message being transmitted are indicated in the first two fields.

The GPGSV log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

> (i) 1. Satellite information may require the transmission of multiple messages. The first field specifies the total number of messages, minimum value 1. The second field identifies the order of this message (message number), minimum value 1.
>
> 2. If the NMEATALKER command, see *page 219*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only) or GL (GLONASS satellites only) or GN (satellites from both systems) or GA (Galileo) or GA (Galileo satellites only). Each system is output in a separate message.
>
> 3. A variable number of 'PRN-Elevation-Azimuth-SNR' sets are allowed up to a maximum of four sets per message. Null fields are not required for unused sets when less than four sets are transmitted.

Message ID: **223**
Log Type: **Synch**

Recommended Input:

    log gpgsv ontime 1

Example (Including GPS and GLONASS sentences):

```
$GPGSV,3,1,11,18,87,050,48,22,56,250,49,21,55,122,49,03,40,284,
47*78
$GPGSV,3,2,11,19,25,314,42,26,24,044,42,24,16,118,43,29,15,039,
42*7E
$GPGSV,3,3,11,09,15,107,44,14,11,196,41,07,03,173,*4D
$GLGSV,2,1,06,65,64,037,41,66,53,269,43,88,39,200,44,74,25,051,*64
$GLGSV,2,2,06,72,16,063,35,67,01,253,*66
```

> (i) The GPGSV log can be used to determine which satellites are currently available to the receiver. Comparing the information from this log to that in the GPGSA log shows if the receiver is tracking all available satellites.
>
> See also the GPGGA usage box that applies to all NMEA logs on *page 474*.

Field	Structure	Field Description	Symbol	Example
1	$GPGSV	Log header		$GPGSV
2	# msgs	Total number of messages (1-9)	x	3
3	msg #	Message number (1-9)	x	1
4	# sats	Total number of satellites in view. May be different than the number of satellites in use (see also the GPSGSA log on *page 474*)	xx	09
5	prn	Satellite PRN number GPS = 1 to 32 SBAS = 33 to 64 (add 87 for PRN#s) GLO = 65 to 96 [a]	xx	03
6	elev	Elevation, degrees, 90 maximum	xx	51
7	azimuth	Azimuth, degrees True, 000 to 359	xxx	140
8	SNR	SNR (C/No) 00-99 dB, null when not tracking	xx	42
... ...	... ...	Next satellite PRN number, elev, azimuth, SNR, ... Last satellite PRN number, elev, azimuth, SNR,		
variable	*xx	Checksum	*hh	*72
variable	[CR][LF]	Sentence terminator		[CR][LF]

a. The NMEA GLONASS PRN numbers are 64 plus the GLONASS slot number. Current slot numbers are 1 to 24 which give the range 65 to 88. PRN numbers 89 to 96 are available if slot numbers above 24 are allocated to on-orbit spares.

3.2.52 GPHDT
NMEA heading log

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains actual vessel heading in degrees True (from True North). See also a description of heading on *page 501*. You can also set a standard deviation threshold for this log, see *page 176.*

> You must have an ALIGN capable receiver to use this log.

> 1. If the NMEATALKER command, see *page 219*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only) or GN (satellites from both systems) or GA (Galileo satellites only).
> 2. Asynchronous logs, such as GPHDT, should only be logged ONCHANGED otherwise the most current data is not available or included in the output. An example of this occurrence is in the ONTIME trigger. If this trigger is not logged ONCHANGED, it may cause inaccurate time tags.

Message ID: **1045**
Log Type: **Asynch**

Recommended Input:

> log gphdt onchanged

Example 1 (GPS only):

`$GPHDT,75.5664,T*36`

Example 2 (Combined GPS and GLONASS):

`$GNHDT,75.5554,T*45`

Field	Structure	Field Description	Symbol	Example
1	$GPHDT	Log header		$GPHDT
2	heading	Heading in degrees	x.x	75.5554
3	True	Degrees True	T	T
4	*xx	Checksum	*hh	*36
5	[CR][LF]	Sentence terminator		[CR][LF]

3.2.53   GPRMB
Navigation information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains navigation data from present position to a destination waypoint. The destination is set active by the receiver SETNAV command (*page 296*).

The GPRMB log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

Message ID: **224**
Log Type: **Synch**

Recommended Input:

> log gprmb ontime 1

Example 1 (GPS only):

```
$GPRMB,A,5.14,L,FROM,TO,5109.7578000,N,11409.0960000,W,5.1,303.0,
-0.0,V,A*6F
```

Example 2 (Combined GPS and GLONASS):

```
$GNRMB,A,5.14,L,FROM,TO,5109.7578000,N,11409.0960000,W,5.1,303.0,
-0.0,V,A*71
```

> (i)   1.   If the NMEATALKER command, see *page 219*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only) or GN (satellites from both systems) or GA (Galileo satellites only).
>
> 2.   See the GPGGA usage box that applies to all NMEA logs on *page 474*.

Table 107: NMEA Positioning System Mode Indicator

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

Field	Structure	Field Description	Symbol	Example
1	$GPRMB	Log header		$GPRMB
2	data status	Data status: A = data valid; V = navigation receiver warning	A	A
3	xtrack	Cross track error [a]	x.x	5.14
4	dir	Direction to steer to get back on track (L/R) [b]	a	L
5	origin ID	Origin waypoint ID [c]	c--c	FROM
6	dest ID	Destination waypoint ID [C]	c--c	TO
7	dest lat	Destination waypoint latitude (DDmm.mm [c]	llll.ll	5109.7578000
8	lat dir	Latitude direction (N = North, S = South) [c]	a	N
9	dest lon	Destination waypoint longitude (DDDmm.mm) [c]	yyyyy.yy	11409.0960000
10	lon dir	Longitude direction (E = East, W = West) [c]	a	W
11	range	Range to destination, nautical miles [d]	x.x	5.1
12	bearing	Bearing to destination, degrees True	x.x	303.0
13	vel	Destination closing velocity, knots	x.x	-0.0
14	arr status	Arrival status: A = perpendicular passed V = destination not reached or passed	A	V
15	mode ind	Positioning system mode indicator, see *Table 107* on *page 492*	a	A
16	*xx	Checksum	*hh	*6F
17	[CR][LF]	Sentence terminator		[CR][LF]

a. - If cross track error exceeds 9.99 NM, display 9.99

   - Represents track error from intended course

   - One nautical mile = 1,852 metres

b. Direction to steer is based on the sign of the crosstrack error, that is, L = xtrack error (+);
   R = xtrack error (-).

c. Fields 5, 6, 7, 8, 9, and 10 are tagged from the SETNAV command, see *page 296*.

d. If range to destination exceeds 999.9 NM, display 999.9.

3.2.54   GPRMC
GPS specific information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains time, date, position, track made good and speed data provided by the GPS navigation receiver. RMC and RMB are the recommended minimum navigation data to be provided by a GNSS receiver.

A comparison of the position precision between this log and other selected NMEA logs can be seen in *Table 106, Position Precision of NMEA Logs* on *page 481.*

The GPRMC log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

> ⓘ If the `NMEATALKER` command, see *page 219*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only) or GN (satellites from both systems) or GA (Galileo satellites only).

Message ID: **225**
Log Type: **Synch**

Recommended Input:

    log gprmc ontime 1

Example 1 (GPS):

```
$GPRMC,144326.00,A,5107.0017737,N,11402.3291611,W,0.080,323.3,
210307,0.0,E,A*20
```

Example 2 (Combined GPS and GLONASS):

```
$GNRMC,143909.00,A,5107.0020216,N,11402.3294835,W,0.036,348.3,
210307,0.0,E,A*31
```

> ⓘ See the GPGGA usage box that applies to all NMEA logs on *page 474*.

Field	Structure	Field Description	Symbol	Example
1	$GPRMC	Log header		$GPRMC
2	utc	UTC of position	hhmmss.ss	144326.00
3	pos status	Position status: A = data valid, V = data invalid	A	A
4	lat	Latitude (DDmm.mm)	llll.ll	5107.0017737

Field	Structure	Field Description	Symbol	Example
5	lat dir	Latitude direction N = North, S = South	a	N
6	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.3291611
7	lon dir	Longitude direction E = East, W = West	a	W
8	speed Kn	Speed over ground, knots	x.x	0.080
9	track true	Track made good, degrees True	x.x	323.3
10	date	Date: dd/mm/yy	xxxxxx	210307
11	mag var	Magnetic variation, degrees [a]	x.x	0.0
12	var dir	Magnetic variation direction E/W [b]	a	E
13	mode ind	Positioning system mode indicator, see *Table 107* on *page 492*	a	A
14	*xx	Checksum	*hh	*20
15	[CR][LF]	Sentence terminator		[CR][LF]

a. Note that this field is the actual magnetic variation and will always be positive. The direction of the magnetic variation is always positive.

b. Easterly variation (E) subtracts from True course and Westerly variation (W) adds to True course.

3.2.55 GPSEPHEM
Decoded GPS ephemerides

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains a single set of GPS ephemeris parameters.

Message ID: 7
Log Type: Asynch

Recommended Input:

> log gpsephema onchanged

ASCII Example:

```
#GPSEPHEMA,COM1,12,59.0,SATTIME,1337,397560.000,00000000,9145,1984;
3,397560.0,0,99,99,1337,1337,403184.0,2.656004220E+07,4.971635660E-
09,-2.752651501E+00,7.1111434372E-03,6.0071892571E-01,2.428889275E-
06,1.024827361E-05,1.64250000E+02,4.81562500E+01,1.117587090E-08,-
7.078051567E-08,9.2668266314E-01,-1.385772009E-10,-2.098534041E+00,
-8.08319384E-09,99,403184.0,-4.190951586E-09,2.88095E-05,3.06954E-
12,0.00000,TRUE,1.458614684E-04,4.00000000E+00*0F875B12
```

```
#GPSEPHEMA,COM1,11,59.0,SATTIME,1337,397560.000,00000000,9145,1984;
25,397560.0,0,184,184,1337,1337,403200.0,2.656128681E+07,4.89734685
1E-09,1.905797220E+00,1.1981436634E-02,-1.440195331E+00,-1.08405947
7E-06,6.748363376E-06,2.37812500E+02,-1.74687500E+01,1.825392246E-
07,-1.210719347E-07,9.5008501632E-01,2.171519024E-10,2.086083072E+
00,-8.06140722E-09,184,403200.0,-7.450580597E-09,1.01652E-04,
9.09495E-13,0.00000,TRUE,1.458511425E-04,4.00000000E+00*18080B24
```

.

.

.

```
#GPSEPHEMA,COM1,0,59.0,SATTIME,1337,397560.000,00000000,9145,1984;1
,397560.0,0,224,224,1337,1337,403200.0,2.656022490E+07,3.881233098E
-09,2.938005195E+00,5.8911956148E-03,-1.716723741E+00,-2.723187208
E-06,9.417533875E-06,2.08687500E+02,-5.25625000E+01,9.126961231E-
08,-7.636845112E-08,9.8482911735E-01,1.325055194E-10,1.162012787
E+00,-7.64138972E-09,480,403200.0,-3.259629011E-09,5.06872E-06,
2.04636E-12,0.00000,TRUE,1.458588731E-04,4.00000000E+00*97058299
```

> (i) The GPSEPHEM log can be used to monitor changes in the orbits of GPS satellites.

Table 108: URA Variance

Index Value (m)	A: Standard Deviations (m)	Variance: A^2 (m^2)
0	2.0	4
1	2.8	7.84
2	4.0	16
3	5.7	32.49
4	8	64
5	11.3	127.69
6	16.0	256
7	32.0	1024
8	64.0	4096
9	128.0	16384
10	256.0	65536
11	512.0	262144
12	1024.0	1048576
13	2048.0	4194304
14	4096.0	16777216
15	8192.0	67108864

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GPSEPHEM header	Log header		H	0
2	PRN	Satellite PRN number	Ulong	4	H
3	tow	Time stamp of subframe 0 (seconds)	Double	8	H+4
4	health	Health status - a 6-bit health code as defined in ICD-GPS-200 [a]	Ulong	4	H+12
5	IODE1	Issue of ephemeris data 1	Ulong	4	H+16
6	IODE2	Issue of ephemeris data 2	Ulong	4	H+20
7	week	GPS reference week number	Ulong	4	H+24
8	z week	Z count week number. This is the week number from subframe 1 of the ephemeris. The 'toe week' (field #7) is derived from this to account for rollover	Ulong	4	H+28
9	toe	Reference time for ephemeris, seconds	Double	8	H+32
10	A	Semi-major axis, metres	Double	8	H+40
11	ΔN	Mean motion difference, radians/second	Double	8	H+48
12	M_0	Mean anomaly of reference time, radians	Double	8	H+56

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
13	ecc	Eccentricity, dimensionless - quantity defined for a conic section where e= 0 is a circle, e = 1 is a parabola, 0<e<1 is an ellipse and e>1 is a hyperbola	Double	8	H+64
14	ω	Argument of perigee, radians - measurement along the orbital path from the ascending node to the point where the SV is closest to the Earth, in the direction of the SV's motion	Double	8	H+72
15	cuc	Argument of latitude (amplitude of cosine, radians)	Double	8	H+80
16	cus	Argument of latitude (amplitude of sine, radians)	Double	8	H+88
17	crc	Orbit radius (amplitude of cosine, metres)	Double	8	H+96
18	crs	Orbit radius (amplitude of sine, metres)	Double	8	H+104
19	cic	Inclination (amplitude of cosine, radians)	Double	8	H+112
20	cis	Inclination (amplitude of sine, radians)	Double	8	H+120
21	I_0	Inclination angle at reference time, radians	Double	8	H+128
22	$\overset{\circ}{I}$	Rate of inclination angle, radians/second	Double	8	H+136
23	ω_0	Right ascension, radians	Double	8	H+144
24	$\overset{\circ}{\omega}$	Rate of right ascension, radians/second	Double	8	H+152
25	iodc	Issue of data clock	Ulong	4	H+160
26	toc	SV clock correction term, seconds	Double	8	H+164
27	tgd	Estimated group delay difference, seconds	Double	8	H+172
28	a_{f0}	Clock aging parameter, seconds (s)	Double	8	H+180
29	a_{f1}	Clock aging parameter, (s/s)	Double	8	H+188
30	a_{f2}	Clock aging parameter, (s/s/s)	Double	8	H+196
31	AS	Anti-spoofing on:0 = FALSE 1 = TRUE	Bool	4	H+204
32	N	Corrected mean motion, radians/second	Double	8	H+208
33	URA	User Range Accuracy variance, m^2. The ICD [a] specifies that the URA index transmitted in the ephemerides can be converted to a nominal standard deviation value using an algorithm listed there. We publish the square of the nominal value (variance). The correspondence between the original URA index and the value output is shown in *Table 108*	Double	8	H+216
34	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+224
35	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. To obtain copies of ICD-GPS-200, refer to ARINC in the Standards and References available on our website at www.novatel.com/support/.

3.2.56   GPVTG
Track made good and ground speed

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the track made good and speed relative to the ground.

The GPVTG log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be one hundred percent accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time status is set to VALID.

Message ID: **226**
Log Type: **Synch**

Recommended Input:

        log gpvtg ontime 1

Example 1 (GPS only):

    $GPVTG,172.516,T,155.295,M,0.049,N,0.090,K,D*2B

Example 2 (Combined GPS and GLONASS):

    $GNVTG,134.395,T,134.395,M,0.019,N,0.035,K,A*33

> If the NMEATALKER command, see *page 219*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only) or GN (satellites from both systems).
>
> See the GPGGA usage box that applies to all NMEA logs on *page 474*.

Field	Structure	Field Description	Symbol	Example
1	$GPVTG	Log header		$GPVTG
2	track true	Track made good, degrees True	x.x	24.168
3	T	True track indicator	T	T
4	track mag	Track made good, degrees Magnetic; Track mag = Track true + (MAGVAR correction) See the *MAGVAR* command, *page 211*	x.x	24.168
5	M	Magnetic track indicator	M	M
6	speed Kn	Speed over ground, knots	x.x	0.4220347
7	N	Nautical speed indicator (N = Knots)	N	N
8	speed Km	Speed, kilometres/hour	x.x	0.781608
9	K	Speed indicator (K = km/hr)	K	K
10	mode ind	Positioning system mode indicator, see *Table 107* on *page 492*	a	A
11	*xx	Checksum	*hh	*7A
12	[CR][LF]	Sentence terminator		[CR][LF]

3.2.57   GPZDA
UTC time and date

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The GPSZDA log outputs the UTC date and time. If no valid almanac is stored in the receiver, a default UTC offset is used to generate the time until a new almanac is downloaded. If the offset is not up-to-date, this initial UTC time may be incorrect until the new almanac is present.

Message ID: **227**
Log Type: **Synch**

Recommended Input:

> log gpzda ontime 1

Example:

```
$GPZDA,143042.00,25,08,2005,,*6E
```

(i)   See the GPGGA usage box that applies to all NMEA logs on *page 474*.

Field	Structure	Field Description	Symbol	Example
1	$GPZDA	Log header		$GPZDA
2	utc	UTC time status	hhmmss.ss	220238.00
3	day	Day, 01 to 31	xx	15
4	month	Month, 01 to 12	xx	07
5	year	Year	xxxx	1992
6	null	Local zone description—not available	xx	(empty when no data is present)
7	null	Local zone minutes description—not available [a]	xx	(empty when no data is present)
8	*xx	Checksum	*hh	*6F
9	[CR][LF]	Sentence terminator		[CR][LF]

a.   Local time zones are not supported by OEM6 family receivers. Fields 6 and 7 are always null.

3.2.58   HEADING
Heading information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The heading is the angle from True North of the base to rover vector in a clockwise direction. This log can be output at both Master and Rover ends.

> ▼ You must have an ALIGN capable receiver to use this log.

> ⓘ Asynchronous logs, such as HEADING, should only be logged ONCHANGED or ONNEW otherwise the most current data is not available or included in the output. An example of this occurrence is in the ONTIME trigger. If this trigger is not logged ONCHANGED, it may cause inaccurate time tags.
>
> The HEADING log is dictated by the output frequency of the master receiver sending out RTCAOBS2, RTCAOBS3 or NovAtelXObs messages. For the OEM628 and OEM638 HEADING now supports 20 Hz output rate. Ensure sufficient radio bandwidth is available between the ALIGN Master and the ALIGN Rover.

Message ID: **971**
Log Type: **Asynch**

Recommended Input:

        log headinga onchanged

ASCII Example:
```
#HEADINGA,COM1,0,77.0,FINESTEERING,1481,418557.000,00000000,3663,
36137;SOL_COMPUTED,L1_INT,5.913998127,75.566444397,-0.152066842,
0.0,0.104981117,0.222061798,"AAAA",13,10,10,0,0,00,0,11*481A5BAB
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	HEADING header	Log header		H	0
2	sol stat	Solution status, see *Table 85* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84* on *page 407*	Enum	4	H+4
4	length	Baseline length (0 to 3000 m)[a] For Z ALIGN Rovers, this field outputs decimal portion of the baseline	Float	4	H+8
5	heading	Heading in degrees (0 to 360.0 degrees)	Float	4	H+12
6	pitch	Pitch (±90 degrees)	Float	4	H+16

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
7	Reserved		Float	4	H+20
8	hdg std dev	Heading standard deviation in degrees	Float	4	H+24
9	ptch std dev	Pitch standard deviation in degrees	Float	4	H+28
10	stn ID	Station ID string	Char[4]	4	H+32
11	#SVs	Number of satellites tracked	Uchar	1	H+36
12	#solnSVs	Number of satellites in solution	Uchar	1	H+37
13	#obs	Number of satellites above the elevation mask angle	Uchar	1	H+38
14	#multi	Number of satellites above the mask angle with L2	Uchar	1	H+39
15	sol source	Solution source (see *Table 109, Solution Source* on *page 502*)	Hex	1	
16	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	
17	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+42
18	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+43
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. This is only true for L1 GPS + GLONASS Heading. If the user has a dual-frequency heading model, traditional RTK baseline lengths apply.

Table 109: Solution Source

Bit	Mask	Description
0-1	0x03	Reserved
2-3	0x0C	Source antenna 0=Primary antenna 1=Secondary antenna
4-7	0xF0	Reserved

3.2.59   HEADING2
Heading information with multiple rovers

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The heading is the angle from True North of the base to rover vector in a clockwise direction. This log is similar to the HEADING log (*page 501*) with additional rover ID field. This log can be output at both Master and Rover ends.

▼ An ALIGN capable receiver is required to use this log.

ⓘ Asynchronous logs, such as HEADING and HEADING2, should only be logged ONCHANGED or ONNEW otherwise the most current data is not available or included in the output. An example of this occurrence is in the ONTIME trigger. If this trigger is not logged ONCHANGED, it may cause inaccurate time tags.

The HEADING2 log is dictated by the output frequency of the master receiver sending out RTCAOBS2, RTCAOBS3 or NovAtelXObs messages. For the OEM628 and OEM638, HEADING2 now supports 20 Hz output rate. Ensure sufficient radio bandwidth is available between the ALIGN Master and the ALIGN Rover.

Message ID: **1335**
Log Type: **Asynch**

Recommended Input:

　　　　log heading2a onnew

ASCII Example:
```
#HEADING2A,COM1,0,39.5,FINESTEERING,1622,422892.200,00040000,F9BF,
6521;SOL_COMPUTED,NARROW_INT,0.927607417,178.347869873,
-1.3037414550.0,0.261901051,0.391376048,"R222","AAAA",18,17,17,
16,0,01,0,33*7BE836F6
```

Field #	Field type	Field Description	Binary Format	Binary Bytes	Binary Offset
1	HEADING2	Log header		H	0
2	sol stat	Solution status, see *Table 85* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84* on *page 407*	Enum	4	H+4
4	length	Baseline length in metres For Z ALIGN Rovers, this field outputs decimal portion of the baseline	Float	4	H+8
5	heading	Heading in degrees (0° to 359.999°)	Float	4	H+12
6	pitch	Pitch (±90 degrees)	Float	4	H+16

Field #	Field type	Field Description	Binary Format	Binary Bytes	Binary Offset
7	Reserved		Float	4	H+32
8	hdg std dev	Heading standard deviation in degrees	Float	4	H+24
9	ptch std dev	Pitch standard deviation in degrees	Float	4	H+28
10	rover stn ID	Rover Receiver ID Set using SETROVERID command on ROVER e.g. setroverid RRRR	Char[4]	4	H+32
11	Master stn ID	Master Receiver ID Set using DGPSTXID command on MASTER (refer to the *Section 2.4.32* on *page 125*) Default: AAAA	Char[4]	4	H+36
12	#SVs	Number of satellites tracked	Uchar	1	H+40
13	#solnSVs	Number of satellites in solution	Uchar	1	H+41
14	#obs	Number of satellites above the elevation mask angle	Uchar	1	H+42
15	#multi	Number of satellites above the mask angle with L2	Uchar	1	H+43
16	sol source	Solution source (see *Table 109, Solution Source* on *page 502*)	Hex	1	
17	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Uchar	1	H+45
18	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+46
19	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+47
20	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+48
21	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.60 HEADINGRATE
Heading rate information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log provides rate of change for the heading parameters. The heading is the angle from True North of the base to rover vector in a clockwise direction.

> ▼ You must have an ALIGN capable receiver to use this log.

Message ID: **1698**
Log Type: **Asynch**

Recommended Input:

log headingratea onchanged

ASCII Example:

610961914,-
0.814819336,0.008199811,0.012566212,0.003494983,0.003570757,0.0,"N04E","0110",0,0,0,0*68A
347B0

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	HEADINGRATE header	Log header		H	0
2	sol stat	Solution status, see *Table 85* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84* on *page 407*	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	length rate	Rate of change of the baseline length in m/s. For Z ALIGN rovers, this field outputs the decimal portion of the baseline rate.	Float	4	H+12
6	heading rate	Rate of change of the heading in °/s	Float	4	H+16
7	pitch rate	Rate of change of the pitch in °/s	Float	4	H+20
8	length rate std dev	Baseline rate standard deviation in m/s	Float	4	H+24
9	heading rate std dev	Heading rate standard deviation in °/s	Float	4	H+28
10	pitch rate std dev	Pitch rate standard deviation in °/s	Float	4	H+32
11	Reserved		Float	4	H+36

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
12	rover stn ID	Rover Receiver ID Set using the *SETROVERID* command on the rover receiver. For example, setroverid RRRR (*page 301*).	Uchar	4	H+40
13	master stn ID	Master Receiver ID Set using the *DGPSTXID* command on the master receiver. Default: AAAA (*page 125*)	Uchar	4	H+44
14	sol source	Solution source (see *Table 109, Solution Source* on *page 502*)	Hex	1	
15	Reserved		Uchar	1	H+49
16	Reserved		Uchar	1	H+50
17	Reserved		Uchar	1	H+51
18	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+52
19	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.61 HEADINGSATS
Satellite used in heading solution

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log provides information on the satellites that are used in a heading solution.

> ⓘ   The HEADINGSATS log can only be used from the ALIGN rover.

Message ID: **1316**
Log Type: **Asynch**

Recommended Input:

       log headingsatsa onnew

ASCII Example:

```
#HEADINGSATSA,COM1,0,26.0,FINESTEERING,1625,344654.600,00000008,F5B
0,6569;17,GPS,31,GOOD,00000003,GPS,23,GOOD,00000003,GPS,30,GOOD,000
00003,GPS,16,GOOD,00000003,GPS,20,GOOD,00000003,GPS,25,GOOD,0000000
3,GPS,4,GOOD,00000003,GPS,24,GOOD,00000003,GPS,11,GOOD,00000003,GPS
,32,GOOD,00000003,GPS,14,GOOD,00000003,GLON-
ASS,20+2,GOOD,00000003,GLONASS,14-7,GOOD,00000001,GLONASS,2-
4,GOOD,00000003,GLONASS,13-2,
GOOD,00000003,GLONASS,12-1,GOOD,00000003,GLONASS,19+3,GOOD,
00000001*15EC53A6
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	HEADINGSATS	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	System	Refer to *Table 110, Satellite System* on *page 508*.	Enum	4	H+4
4	Satellite ID	In binary logs, the satellite ID field is 4 bytes. The 2 lowest-order bytes, interpreted as a USHORT, are the system identifier: for instance, the PRN for GPS, or the slot for GLONASS. The 2 highest-order bytes are the frequency channel for GLONASS, interpreted as a SHORT and zero for all other systems. In ASCII and abbreviated ASCII logs, the satellite ID field is the system identifier. If the system is GLONASS and the frequency channel is not zero, then the signed channel is appended to the system identifier. For example, slot 13, frequency channel -2 is output as 13-2	Ulong	4	H+8
5	Status	see *Table 90, Observation Statuses* on *page 413*	Enum	4	H+12
6	Signal Mask	see *Table 91* through *Table 94*	Hex	4	H+16
7	Next satellite offset = H + 4 + (#sat x 16)				

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+(# satx16)
9	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Table 110: Satellite System

Binary Value	ASCII Mode Name
0	GPS
1	GLONASS
2	SBAS
5	Galileo
6	BeiDou
7	QZSS

3.2.62 HWMONITOR
Monitor hardware levels

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log allows the user to monitor temperature, antenna current and voltages.

Message ID: 963
Log Type: Polled

Recommended Input:

log hwmonitora ontime 10

ASCII Example:

```
#HWMONITORA,COM1,0,83.0,FINESTEERING,1681,319572.841,00000020,52DB,
10526;
10,
43.312500000,100,0.000000000,200,3.207038403,300,1.881818175,400,2.
787878752,500,5.085718632,600,0.000000000,700,1.200586438,800,1.313
000917,900,1.816617727,A00*FE36FDDC
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	HWMONITOR header	Log Header		H	0
2	# measurements	Number of measurements to follow	Ulong	4	H
3	reading	Temperature, antenna current or voltage reading Units: Degree Celsius for Temperature Amps for Antenna Current Volts for Voltage	Float	4	H+4
4	status	See *Table 111*	HexUlong	4	H+8
5...	Next reading offset = H + 4 + (# measurements x 8)				
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (# measurements x 8)
7	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Table 111: HWMONITOR Status Table

Bits	Description	Applicable OEM Platform

0-7	Boundary Limit Status (Binary):
	   0 = Value falls within acceptable bounds
	   1 = Value is under the lower warning limit
	   2 = Value is under the lower error limit
	   3 = Value is over the upper warning limit
	   4 = Value is over the upper error limit	

8-15	Reading Type (Binary): 0 = Reserved	
	1 = Temperature	615, 617, 628, 638
	2 = Antenna Current	628, 638
	3 = MID3V3 Voltage	628, 638
	4 = VNIOL Voltage	628
	5 = VNIOH Voltage	628
	6 = Supply Voltage	628, 638
	7 = Antenna Current Voltage	628
	8 = MINOS Core Voltage	628, 638
	9 = VCC Core Voltage	628
	10 = VCC Mem Voltage	628
	11 = Atlas Voltage	628
	12 = 3V3 Imon	None
	13 = GPIA	638
	14 = FPGA Voltage	638
	15 = MINOS IO	638
	16 = VIOL	638
	17 = 1V8	638
	18 = VDD1	638
	19 = VDD2	638
	20 = LNA IMON	638
	21 = 5 V	638
	22 = Secondary Temperature	638

3.2.63 IONUTC
Ionospheric and UTC data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the Ionospheric Model parameters (ION) and the Universal Time Coordinated parameters (UTC).

Message ID: **8**
Log Type: **Asynch**

Recommended Input:

> log ionutca onchanged

ASCII Example:

```
#IONUTCA,COM1,0,58.5,FINESTEERING,1337,397740.107,00000000,EC21,
1984;1.210719347000122E-08,2.235174179077148E-08,
-5.960464477539062E-08,-1.192092895507812E-07,
1.003520000000000E+05,1.146880000000000E+05,
-6.553600000000000E+04,-3.276800000000000E+05,1337,589824,
-1.2107193470001221E-08,-3.907985047E-14,1355,7,13,14,0*C1DFD456
```

> (i) The Receiver-Independent Exchange (RINEX1[a]) format is a broadly accepted, receiver independent format for storing GPS data. It features a non-proprietary ASCII file format that can be used to combine or process data generated by receivers made by different manufacturers.
>
> Use the NovAtel's Convert4 utility to produce RINEX files from NovAtel receiver data files. For the best results, the NovAtel receiver input data file should contain the logs as specified in the *PC Software and Firmware* chapter of the OEM6 Family Installation and Operation User Manual (OM-20000128) including IONUTC.

> a. Refer to the U.S. National Geodetic Survey website at: www.ngs.noaa.gov/CORS/data.shtml.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	IONUTC header	Log header		H	0
2	a0	Alpha parameter constant term	Double	8	H
3	a1	Alpha parameter 1st order term	Double	8	H+8
4	a2	Alpha parameter 2nd order term	Double	8	H+16
5	a3	Alpha parameter 3rd order term	Double	8	H+24
6	b0	Beta parameter constant term	Double	8	H+32
7	b1	Beta parameter 1st order term	Double	8	H+40
8	b2	Beta parameter 2nd order term	Double	8	H+48

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
9	b3	Beta parameter 3rd order term	Double	8	H+56
10	utc wn	UTC reference week number	Ulong	4	H+64
11	tot	Reference time of UTC parameters	Ulong	4	H+68
12	A0	UTC constant term of polynomial	Double	8	H+72
13	A1	UTC 1st order term of polynomial	Double	8	H+80
14	wn lsf	Future week number	Ulong	4	H+88
15	dn	Day number (the range is 1 to 7 where Sunday = 1 and Saturday = 7)	Ulong	4	H+92
16	deltat ls	Delta time due to leap seconds	Long	4	H+96
17	deltat lsf	Future delta time due to leap seconds	Long	4	H+100
18	Reserved (4 bytes, with offset H+104)				
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+108
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.64 IPSTATS
IP statistics

OEM Platform: 628, 638, FlexPak6, ProPak6

This log contains the current IP interface statistics

Message ID: **1669**
Log Type: **Polled**

Recommended Input:

 log ipstatsa

ASCII Example:
```
#IPSTATSA,COM1,0,70.5,FINESTEERING,1749,328376.337,00000020,0D94,
45068;1,CELL,0,526,526*01C4847C
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	IPSTATS header	Log Header		H	0
2	Physical Interface	IP Interface Type 1 = ALL 2 = ETHA 10 = WIFI 20 = CELL	Enum	4	H
3	Reserved		Ulong	4	H+4
4	Receive Bytes	Total number of bytes received	Ulong	4	H+8
5	Transmit Bytes	Total number of bytes transmitted	Ulong	4	H+12
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.65　IPSTATUS
Current network configuration status

OEM Platform: 628, 638, FlexPak6, ProPak6

This log provides the configuration of IP address, netmask, gateway and a list of DNS servers currently in use.

Message ID:　　　　**1289**
Log Type:　　　　　**Asynch**

Recommended Input:

　　　　　log ipstatusa once

ASCII Example:

```
#IPSTATUSA,COM1,0,90.5,FINESTEERING,1609,500464.121,00000000,7FE2,
6259;1,ETHA,"10.4.44.131","255.255.255.0","10.4.44.1",1,
"198.161.72.85"*EC22236C
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	IPSTATUS Header	Log header	-	H	0
2	#IPrec	Number of records to follow	Ulong	4	H
3	interface	Name of the network interface 2 = ETHA 10 = WIFI 20 = CELL	Enum	4	H+4
4	IP address	IP Address-decimal dot notation	String [16]	variable[a]	H+8
5	netmask	Netmask-decimal dot notation	String [16]	variable[a]	H+24
6	gateway	Gateway-decimal dot notation The gateway address is only included in the IP record for the network interface that is configured as the preferred network interface. This is the default gateway that is currently in use by the receiver (see the SETPREFERREDNETIF command on *page 299*).	String [16]	variable[a]	H+40
7...	Next reading offset = H+4+(#IPrec * 52)				
8	#dnsserver	Number of DNS Servers to follow The DNS servers used by the receiver are restricted to those DNS servers configured for the preferred network interface (see the SETPREFERREDNETIF command on *page 299*).	Ulong	4	H+4+ (#IPrec x 52)
9	server IP address	IP address-decimal dot notation	String [16]	variable[a]	H+4+ (#IPrec x 52)+4
10...	Next reading offset = H+4+(#IPrec * 52)+4+(#dnsserver * 16)				

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
11	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#IPrec x 52)+4+ (#dnsserver x 16)
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

3.2.66   LBANDBEAMTABLE
List of L-Band beams

OEM Platform: 628, 638, FlexPak6, ProPak6

This log lists the TerraStar and Veripos L-Band beams known to the receiver.

Message ID: **1718**
Log Type: **Asynch**

Recommended Input:

>   log lbandbeamtablea onchanged

Abbreviated ASCII Example:

```
<LBANDBEAMTABLE COM1 0 74.5 UNKNOWN 0 0.151 00440000 F3B2 45228
<     7
<          "AORE" "A" 1539982500 1200 -15.50 1
<          "AORW" "B" 1539892500 1200 -54.00 1
<          "IOR" "C" 1539902500 1200 64.50 1
<          "POR" "D" 1539942500 1200 178.00 1
<          "25E" "E" 1539882500 1200 25.00 1
<          "143.5E" "F" 1539992500 1200 143.50 1
<          "98W" "G" 1539902500 1200 -98.00 1
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	LBAND BEAM TABLE header	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	Name	Beam/transmitting satellite name	Char[8]	8	H+4
4	Region ID	Region ID	Char[8]	8	H+12
5	Frequency	Frequency (Hz)	Ulong	4	H+20
6	Baud	Baud rate	Ulong	4	H+24
7	Longitude	Transmitting satellite longitude (degrees)	Float	4	H+28
8	Reserved		Ulong	4	H+32
9	Next beam offset = H + 4 + (#entries x 32)				
10	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#entries x 32)
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.67 LBANDINFO
L-Band configuration information

OEM Platform: 628, 638, FlexPak6, ProPak6

This log outputs configuration information for OmniSTAR services.

> ⓘ In addition to a NovAtel receiver with L-Band capability, a subscription to an OmniSTAR service is required. Contact NovAtel for details through our website at www.novatel.com through *Support* or the *Customer Support* section in the OEM6 Family Installation and Operation User Manual (OM-20000128).

Message ID: 730
Log Type: Asynch

Recommended Input:

       log lbandinfoa ontime 1

ASCII Example 1 (OmniSTAR HP):

```
#LBANDINFOA,COM2,0,81.5,FINESTEERING,1295,152639.184,00000240,C51D,
34461;1547547,4800,C685,0,762640,EXPIRED,0,0,FIXEDTIME,1199,259199,
0*8CC5E573
```

Table 112: L-Band Subscription Type

Binary	ASCII	Description
0	EXPIRED	The L-Band subscription has expired or does not exist
1	FIXEDTIME	The L-Band subscription expires at a fixed date and time
2	COUNTDOWN	The L-Band subscription expires after the specified amount of running time
3	COUNTDOWNOVERRUN	The COUNTDOWN subscription has expired but has entered a brief grace period. Resubscribe immediately
16	UNKNOWN	Unknown subscription

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	LBANDINFO header	Log header		H	0
2	freq	Selected frequency for L-Band service (Hz or kHz, depending on how the frequency was specified in the ASSIGNLBAND, ASSIGNLBAND2 or ASSIGNLBANDBEAM command)	Ulong	4	H
3	baud	Communication baud rate from L-Band satellite	Ulong	4	H+4
4	ID	L-Band signal service ID	Ushort	2	H+8
5			Ushort	2	H+10

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
6	OSN	L-Band serial number	Ulong	4	H+12
7	vbs sub	L-Band VBS subscription type (see *Table 112* on *page 517*)	Enum	4	H+16
8	vbs exp week	GPS reference week number of L-Band VBS expiration date [a]	Ulong	4	H+20
9	vbs exp secs	Number of seconds into the GPS reference week of L-Band VBS expiration date [a]	Ulong	4	H+24
10	hp sub	OmniSTAR HP or XP subscription type (see *Table 112* on *page 517*)	Enum	4	H+28
11	hp exp week	GPS reference week number of OmniSTAR HP or XP expiration date [a]	Ulong	4	H+32
12	hp exp secs	Number of seconds into the GPS reference week of OmniSTAR HP or XP expiration date [a]	Ulong	4	H+36
13	hp sub mode	Subscription mode if the subscription is valid. Refer to *Table 113* on *page 518*	Hex	4	H+40
14	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. If the subscription type is COUNTDOWN, see Field #7 above, the expiration week and expiration seconds into the GPS reference week contain the amount of running time remaining in the subscription.

If the subscription type is COUNTDOWNOVERRUN, the expiration week and expiration seconds into GPS reference week count the amount of the overrun time.

Table 113: HP Subscription Mode

Bit Number	Mask	Description[a]	Value
0	0x00000001	XP	1=subscribed
1	0x00000002	HP	1=subscribed
2	0x00000004	G2	1=subscribed
3-7	0x000000f8	Reserved	
9	0x00000100	GPS measurements	1=enabled
10	0x00000200	GLONASS measurements	1=enabled

a. All other bits are reserved.

3.2.68   LBANDSTAT
L-Band status information

OEM Platform: 628, 638, FlexPak6, ProPak6

This log outputs status information for a standard L-Band, OmniSTAR XP (Extra Precision) or OmniSTAR HP (High Performance) service.

> ⓘ In addition to a NovAtel receiver with L-Band capability, a subscription to an OmniSTAR service is required. Contact NovAtel for details, through our website at www.novatel.com through *Support*, or the *Customer Support* section in the OEM6 Family Installation and Operation User Manual.

Message ID: **731**
Log Type: **Synch**

Recommended Input:

      log lbandstata ontime 1

ASCII Example:

```
#LBANDSTATA,COM1,0,73.5,FINESTEERING,1314,494510.000,00000000,C797,
1846;1551488896,43.19,62.3,0.00,0082,0000,7235,11,0,0000,0001,7762,
04000000,0*93F7D2AF
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	LBANDSTAT header	Log header		H	0
2	freq	Measured frequency of L-Band signal (Hz)	Ulong	4	H
3	C/No	Carrier to noise density ratio C/No = $10[\log_{10}(S/N_0)]$ (dB-Hz)	Float	4	H+4
4	locktime	Number of seconds of continuous tracking (no cycle slipping)	Float	4	H+8
5	Reserved		Float	4	H+12
6	tracking	Tracking status of L-Band signal (see *Table 114* on *page 520*)	Hex	2	H+16
7	Reserved		Hex	2	H+18
8	#bytes	Number of bytes fed to the standard process	Ulong	4	H+20
9	#good dgps	Number of standard updates	Ulong	4	H+24
10	#bad data	Number of missing standard updates	Ulong	4	H+28
11	Reserved (the *hp status 1* field is obsolete and has been replaced by the longer OmniSTAR HP Status field. The shorter legacy status here is maintained for backward compatibility)		Hex	2	H+32

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
12	hp status 2	Additional status pertaining to the HP or XP process (see *Table 115* on *page 521*)	Hex	2	H+34
13	#bytes hp	Number of bytes fed to the HP or XP process	Ulong	4	H+36
14	hp status	Status from the HP or XP process (see *Table 116* on *page 521*)	Hex	4	H+40
15	VBS status	Status word for OmniSTAR (see *Table 117* on *page 522*)	Hex	4	H+44
16	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+48
17	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 114: L-Band Signal Tracking Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x0001	Tracking State	0 = Searching, 1 = Pull-in, 2 = Tracking, 3 = Idle
N0	1	0x0002	Tracking State	0 = Searching, 1 = Pull-in, 2 = Tracking, 3 = Idle
N0	2	0x0004	Reserved	
N0	3	0x0008	Reserved	
N1	4	0x0010	Reserved	
N1	5	0x0020	Reserved	
N1	6	0x0040	Bit Timing Lock	0 = Not Locked, 1 = Locked
N1	7	0x0080	Phase Locked	0 = Not Locked, 1 = Locked
N2	8	0x0100	DC Offset Unlocked	0 = Good, 1 = Warning
N2	9	0x0200	AGC Unlocked	0 = Good, 1 = Warning
N2	10	0x0400	Reserved	
N2	11	0x0800	Reserved	
N3	12	0x1000	Reserved	
N3	13	0x2000	Reserved	
N3	14	0x4000	Reserved	
N3	15	0x8000	Error	0 = Good, 1 = Error

Table 115: OmniSTAR HP/XP/G2 Additional Status Word

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x0001	Solution not fully converged	False	True
	1	0x0002	OmniSTAR satellite list available	False	True
	2	0x0004	Reserved		
	3	0x0008			
N1	4	0x0010	HP not authorized [a]	Authorized	Unauthorized
	5	0x0020	XP not authorized [a]	Authorized	Unauthorized
	6	0x0040			
	7	0x0080			
N2	8	0x0100			
	9	0x0200			
	10	0x0400	Reserved		
	11	0x0800			
N3	12	0x1000			
	13	0x2000			
	14	0x4000			
	15	0x8000			

a. This authorization is related to the receiver model and not the OmniSTAR subscription. To view OmniSTAR subscription information use the LBANDINFO log, see *page 517*.

Table 116: OmniSTAR HP/XP/G2 Status Word

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Subscription Expired [a]	False	True
	1	0x00000002	Out of Region [a]	False	True
	2	0x00000004	Wet Error [a]	False	True
	3	0x00000008	Link Error [a]	False	True
N1	4	0x00000010	No Measurements	False	True
	5	0x00000020	No Ephemeris	False	True
	6	0x00000040	No Initial Position	False	True
	7	0x00000080	No Time Set	False	True

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N2	8	0x00000100	Velocity Error	False	True
	9	0x00000200	No base stations	False	True
	10	0x00000400	No Mapping Message	False	True
	11	0x00000800	Altitude Error	False	True
N3-N5	12-23	Reserved			
N6	24-25				
	26	0x04000000	Static Initialization Mode	False	True
	27	Reserved			
N7	28-30				
	31	0x80000000	Updating Data	False	True

a. Contact OmniSTAR for subscription support. All other status values are updated by collecting the OmniSTAR data for 20-35 minutes.

Table 117: OmniSTAR VBS Status Word

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Subscription Expired [a]	False	True
	1	0x00000002	Out of Region [a]	False	True
	2	0x00000004	Wet Error [a]	False	True
	3	0x00000008	Link Error [a]	False	True
N1	4	0x00000010	No Remote Sites	False	True
	5	0x00000020	No Almanac	False	True
	6	0x00000040	No Position	False	True
	7	0x00000080	No Time	False	True
N2-N4	8-19	Reserved			
N5	20				
	21				
	22				
	23	0x00800000	Wet Warning [a]	False	True
N6	24-27	Reserved			
N7	28				
	29				
	30				
	31	0x80000000	Updating Data	False	True

a. Contact OmniSTAR for subscription support. All other status values are updated by collecting OmniSTAR data for 20-35 minutes.

3.2.69 LBANDTRACKSTAT
L-Band tracking status

OEM Platform: 628, 638, FlexPak6, ProPak6

This log reports the L-Band tracking and Viterbi decoding status.

Message ID: 1201
Log Type: Synch

Recommended Input:

log lbandtrackstata ontime 1

ASCII Example:

```
#LBANDTRACKSTATA,COM1,0,73.5,FINESTEERING,
1769,328196.000,00000000,29FD,12602;1,"98W",1539902500,1200,974C,00
C2,0,-316.186,43.842,4.3840,61.920,1088,2,2,
138176,79,0.0001*3E43CB7D
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	LBAND TRACKST AT header	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	Name	Beam/transmitting satellite name	Char[8]	8	H+4
4	Frequency	Frequency assigned to this L-Band beam (Hz)	Ulong	4	H+12
5	Baud rate	Baud rate of assigned beam	UShort	2	H+16
6	ID	Service ID of the assigned beam	UShort	2	H+18
7	Status	Tracking status word (refer to *Table 114, L-Band Signal Tracking Status* on *page 520*)	UShort	2	H+20
8	Reserved	Reserved	UShort	2	H+22
8	Doppler	Signal Doppler (Hz)	Float	4	H+24
9	C/No	Carrier to noise density ratio (dB-Hz)	Float	4	H+28
10	Std. Dev.	Phase error standard deviation (cycles)	Float	4	H+32
11	Lock time	Lock time (seconds)	Float	4	H+36
12	Unique word bits	Total unique word bits	Ulong	4	H+40
13	Bad unique word bits	Bad unique word bits	Ulong	4	H+44
14	Bad unique words	Bad unique words	Ulong	4	H+48
15	Viterbi symbols	Total Viterbi symbols	Ulong	4	H+52

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
16	Corrected Viterbi	Corrected Viterbi symbols	Ulong	4	H+56
17	BER	Estimated pre-Viterbi Bit Error Rate (BER)	Float	4	H+60
18	Next entry offset = H + 4 + (#entries x 56)				
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#entries x 60)
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.70 LOGFILESTATUS
Current state of file and recording

OEM Platform: OEM638, ProPak6

This log displays the current state of the data log file and recording media. Typically it is used to determine if the logfile used to record data to internal storage is open for writing or closed. However, it can also indicate if the internal storage device is busy (for example during a DOSCMD function) or if there's an error with the device.

Message ID: **1146**
Log Type: **Asynch**

Recommended Input:

> log logfilestatus

> (i) The following example indicates that the log file is closed (thus no filename and size), media capacity is 3.8 GB and space remaining is 3.8 GB.

ASCII Example:
```
#LOGFILESTATUSA,COM1,0,82.0,COARSESTEERING,1738,399162.010,00400020
,820B,44894;CLOSE,"",0,INTERNAL_FLASH,3809988,3809996*76EC6431
```

Field	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	LOGFILESTATUS header	Log Header	Header	H	0
2	File State	File/Media State [a] 0 = Open 1 = Close 2 = Media Busy 3 = Media Error	Enum	4	H
3	File Name	File name	Char array	128	H+4
4	File Size	File Size (in Bytes)	Ulong	4	H+132
5	Storage Medium	Recording Device 4 = Internal Flash	Enum	4	H+136
6	Media Free Space	Media Remaining Capacity (in kB)	Ulong	4	H+140
7	Media Capacity	Media Total Capacity (in kB)	Ulong	4	H+144
8	[CR][LF]	Sentence terminator	-	-	-

a. In normal conditions, the status is Open or Close. At start up, during system initialization or in the case of actual media error, the status will be Busy or Error.

3.2.71 LOGLIST
List of system logs

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log outputs a list of log entries in the system. The following tables show the binary and ASCII output. See also the RXCONFIG log on *page 657* for a list of current command settings.

Message ID: **5**
Log Type: **Polled**

Recommended Input:

> log loglista once

ASCII Example:

```
#LOGLISTA,COM1,0,60.5,FINESTEER-
ING,1337,398279.996,00000000,C00C,1984; 8,
COM1,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
COM2,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
COM3,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
USB1,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
USB2,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
USB3,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
COM1,BESTPOSA,ONTIME,10.000000,0.000000,NOHOLD,
COM1,LOGLISTA,ONCE,0.000000,0.000000,NOHOLD*5B29EED3
```

> 🛑 Do not use undocumented logs or commands. Doing so may produce errors and void your warranty.

Binary

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	LOGLIST (binary) header	Log header		H	0
2	#logs	Number of messages to follow, maximum = 64	Ulong	4	H
3	port	Output port, see *Table 4, Page 27*	Enum	4	H+4
4	message	Message ID of log (refer *Table 81, OEM6 Logs in Numerical Order* on *page 375*)	Ushort	2	H+8

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset

5	message type	Bits 0-4 = Reserved
		Bits 5-6 = Format
		    00 = Binary
		    01 = ASCII
		    10 = Abbreviated ASCII,
		        NMEA
		    11 = Reserved
		Bit 7 = Response Bit (see *Section 1.2, Responses* on *page 32*)
		    0 = Original Message
		    1 = Response Message
			Char	1	H+10

6	Reserved		Char	1	H+11
7	trigger	0 = ONNEW 1 = ONCHANGED 2 = ONTIME 3 = ONNEXT 4 = ONCE 5 = ONMARK	Enum	4	H+12
8	period	Log period for ONTIME	Double	8	H+16
9	offset	Offset for period (ONTIME trigger)	Double	8	H+24
10	hold	0 = NOHOLD 1 = HOLD	Enum	4	H+32
11...	Next log offset = H + 4 + (#logs x 32)				
variable	xxxx	32-bit CRC	Hex	4	H+4+ (#logs x 32)

ASCII

Field #	Field type	Data Description	Format
1	LOGLIST (ASCII) header	Log header	
2	#port	Number of messages to follow, maximum = 64	Long
3	port	Output port, see *Table 4, Page 27*	Enum
4	message	Message name of log with no suffix for abbreviated ascii, an A suffix for ascii and a B suffix for binary (*Table 80, OEM6 Logs in Alphabetical Order* on *page 367*)	Char [ ]
5	trigger	ONNEW ONCHANGED ONTIME ONNEXT ONCE ONMARK	Enum
6	period	Log period for ONTIME	Double

Field #	Field type	Data Description	Format
7	offset	Offset for period (ONTIME trigger)	Double
8	hold	NOHOLD HOLD	Enum
9...	Next port		
variable	xxxx	32-bit CRC	Hex
variable	[CR][LF]	Sentence terminator	-

3.2.72 MARK1COUNT
Count for Mark1 input

OEM Platform: 638, ProPak6

> (i) This command 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 147*, the MARKxCOUNT logs become available.

> (i) 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,FINESTEER-
ING,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 (microseconds)	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)	-	-	-

3.2.73 MARK2COUNT
Count for Mark2 input

OEM Platform: 638, ProPak6

> (i) This command 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 147*, the MARKxCOUNT logs become available.

> (i) 1. Use the ONNEW trigger with this, the MARKxTIME, or the MARKxPVA logs.
>
> 2. Only the MARKxCOUNT, MARKxPVA logs, the MARKxTIME 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,FINESTEER-
ING,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 (microseconds)	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)	-	-	-

3.2.74   MARK3COUNT
Count for Mark3 input

OEM Platform: 638, ProPak6

> (i) This command 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 147*, the MARKxCOUNT logs become available.

> (i) 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,FINESTEER-
ING,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 (microseconds)	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)	-	-	-

3.2.75   MARK4COUNT
Count for Mark4 input

OEM Platform: 638, ProPak6

> (i) This command 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 149*, the MARKxCOUNT logs become available.

> (i) 1. Use the ONNEW trigger with this, the MARKxTIME, or the MARKxPVA logs.
>
> 2. Only the MARKxCOUNT, MARKxPVA logs, the MARKxTIME 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,FINESTEER-
ING,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 (microseconds)	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)	-	-	-

3.2.76 MARKPOS, MARK2POS, MARK3POS and MARK4POS Position at time of mark input event

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the estimated position of the antenna when a pulse is detected at a mark input. MARKPOS is generated when a pulse occurs on the MK1I input. MARK2POS is generated when a pulse occurs on the MK2I input.MARK3POS is generated when a pulse occurs on the MK3I input (OEM638 and ProPak6 only). MARK4POS is generated when a pulse occurs on the MK4I input (OEM638 and ProPak6 only). Refer to the product specific *Technical Specifications* appendices in the OEM6 Family Installation and Operation User Manual (OM-20000128) for mark input pulse specifications and the location of the mark input pins.

The position at the mark input pulse is extrapolated using the last valid position and velocities. The latched time of mark impulse is in GPS reference weeks and seconds into the week. The resolution of the latched time is 10 ns. See also the notes on MARKPOS in the MARKTIME log on *page 536*.

Message ID:	**181 (MARKPOS)**
	615 (MARK2POS)
	1738 (MARK3POS)
	1739 (MARK4POS)

Log Type: **Asynch**

Recommended Input:

    log markposa onnew
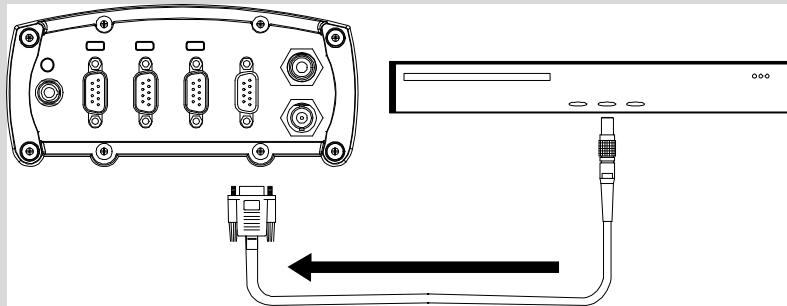
ⓘ
1. Use the ONNEW trigger with the MARKTIME or MARKPOS logs.
2. Refer to the Technical Specifications appendix in the OEM6 Family Installation and Operation User Manual (OM-20000128) for more details on the MK1I pin. ONMARK only applies to MK1I. Events on MK2I (if available) do not trigger logs when ONMARK is used. Use the ONNEW trigger with the MARKTIME, MARK2TIME, MARKPOS or MARK2POS logs.
3. b.Once the 1PPS signal has hit a rising edge, for both MARKPOS and MARKTIME logs, a resolution of both measurements is 10 ns. As for the ONMARK trigger for other logs that measure latency, for example RANGE and POSITION logs such as BESTPOS, it takes typically 20-30 ms (50 ms maximum) for the logs to output information from the 1PPS signal. Latency is the time between the reception of the 1PPS pulse and the first byte of the associated log. See also the MARKPOS and MARKTIME logs starting on *Page 533*.

Abbreviated ASCII Example:
```
<MARKPOS COM1 0 89.0 FINESTEERING 1670 413138.000 00000020 C223 42770
< SOL_COMPUTED SINGLE 51.11289233689 -114.02932170726 1018.9653
1049.4915 BUKIT 1.9372 1.1981 4.0909 "" 0.000 0.000 19 18 18 18 0 06
0 33
```

> ⓘ Consider the case where you have a user point device such as video equipment. Connect the device to the receiver's I/O port using a cable that is compatible to both the receiver and the device. Refer to your device's documentation for information about connectors and cables. The arrow along the cable in the figure below indicates a MARKIN pulse, from the user device on the right to the receiver I/O port.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	MARKPOS/ MARK2POS/ MARK3POS/ MARK4POS header	Log header		H	0
2	sol status	Solution status (see *Table 85* on *page 408*)	Enum	4	H
3	pos type	Position type (see *Table 84* on *page 407*)	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 (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m) [a]	Float	4	H+32
8	datum id#	Datum ID number (refer to *Table 23, Reference Ellipsoid Constants* on *page 119*)	Enum	4	H+36
9	lat σ	Latitude standard deviation (m)	Float	4	H+40
10	lon σ	Longitude standard deviation (m)	Float	4	H+44
11	hgt σ	Height standard deviation (m)	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 satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+65

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
17	#ggL1	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 410*)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	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.

3.2.77 MARKTIME, MARK2TIME, MARK3TIME, MARK4TIME
Time of mark input event

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the time of the leading edge of the detected mark input pulse.

- MARKTIME is generated when a pulse occurs on a MK1I input
- MARK2TIME is generated when a pulse occurs on a MK2I input
- MARK3TIME is generated when a pulse occurs on a MK3I input (OEM638 and ProPak6 only)
- MARK4TIME is generated when a pulse occurs on a MK4I input (OEM638 and ProPak6 only)

Refer to the *Technical Specifications* appendix in the OEM6 Family Installation and Operation User Manual (OM-20000128) for mark input pulse specifications and the location of the mark input pins. The resolution of this measurement is 10 ns.

> (i)
> 1. Use the ONNEW trigger with the MARKTIME or the MARKPOS logs.
> 2. Only the MARKPOS logs, MARKTIME 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.
> 3. Refer to the Technical Specifications appendix in the OEM6 Family Installation and Operation User Manual (OM-20000128) for more details on the MK1I pin. ONMARK only applies to MK1I. Events on MK2I (if available) do not trigger logs when ONMARK is used. Use the ONNEW trigger with the MARKTIME, MARK2TIME, MARKPOS or MARK2POS logs.
> 4. b.Once the 1PPS signal has hit a rising edge, for both MARKPOS and MARKTIME logs, a resolution of both measurements is 10 ns. As for the ONMARK trigger for other logs that measure latency, for example RANGE and POSITION logs such as BESTPOS, it takes typically 20-30 ms (50 ms maximum) for the logs to output information from the 1PPS signal. Latency is the time between the reception of the 1PPS pulse and the first byte of the associated log. See also the MARKPOS and MARKTIME logs starting on *Page 533*.

Message ID:	**231**	**(MARKTIME)**
	616	**(MARK2TIME)**
	1075	**(MARK3TIME)**
	1076	**(MARK4TIME)**
Log Type:	**Asynch**	

Recommended Input:

    log marktimea onnew

ASCII Example:
```
#MARKTIMEA,COM1,0,77.5,FINESTEERING,1358,422621.000,00000000,292E,
2214;1358,422621.000000500,-1.398163614E-08,7.812745577E-08,
-14.000000002,VALID*D8502226
```

> (i) These logs allow you to measure the time when events are occurring in other devices (such as a video recorder). See also the `MARKCONTROL` command on *page 214*.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	MARKTIME/ MARK2TIME/ MARK3TIME/ MARK4TIME header	Log header	H		0
2	week	GPS reference week number	Long	4	H
3	seconds	Seconds into the week as measured from the receiver clock, coincident with the time of electrical closure on the Mark Input port	Double	8	H+4
4	offset	Receiver clock offset, in seconds. A positive offset implies that the receiver clock is ahead of GPS reference time. To derive GPS reference time, use the following formula: GPS reference time = receiver time - (offset)	Double	8	H+12
5	offset std	Standard deviation of receiver clock offset (s)	Double	8	H+20
6	utc offset	This field represents the offset of GPS reference time from UTC time (s), computed using almanac parameters. UTC time is GPS reference time plus the current UTC offset plus the receiver clock offset. UTC time = GPS reference time + offset + UTC offset[a]	Double	8	H+28
7	status	Clock model status, see *Table 99, Clock Model Status* on *page 433*	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. 0 indicates that UTC time is unknown because there is no almanac available in order to acquire the UTC offset.

3.2.78 MASTERPOS
Master Position using ALIGN

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

ALIGN generates distance and bearing information between a Master and Rover receiver. This log outputs the position information of the master when using the ALIGN feature. This log can be output from both Y and Z ALIGN models and can be output at both Master and Rover ends.

> You must have an ALIGN capable receiver to use this log.

> 1. ALIGN is useful for obtaining the relative directional heading of a vessel/body, separation heading between two vessels/bodies or heading information with moving base and pointing applications.
>
> 2. The log can be output at both Y and Z model Rover if it is receiving the RTCAREFEXT or NovAtelXRef message from the Master. The log can be output at any Master if Master is receiving HEADINGEXTB or HEADING-EXT2B from the Rover. Refer to the NovAtel application note APN-048 for details on HEADINGEXT (available on our website at www.nova-tel.com/support/.)
>
> 3. MASTERPOS logging is dictated by the output frequency of the RTCAREFEXT or NovAtelXARef output frequency.

Message ID: **1051**
Log Type: **Asynch**

Recommended Input:

     log masterposa onchanged

ASCII Example:

```
#MASTERPOSA,COM1,0,21.5,FINESTEERING,1544,340322.000,00000008,5009,
4655;SOL_COMPUTED,NARROW_INT,51.11604599076,-114.03855412002,
1055.7756, 16.9000,WGS84,0.0090,0.0086,0.0143,"AAAA",0.0,0.0,13,
13,13,12,0,0,0,0*A72E8D3F
```

> Asynchronous logs, such as MASTERPOS, should only be logged ONCHANGED or ONNEW otherwise, the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

Field #	Field Type	Field Description	Format	Binary Bytes	Binary Offset
1	MASTERPOS header	Log Header		H	0
2	sol stat	Solution Status, see *Table 85* on *page 408*	Enum	4	H
3	pos type	Position Type see *Table 84* on *page 407*	Enum	4	H+4
4	lat	Master WGS84 Latitude in degrees	Double	8	H+8
5	long	Master WGS84 Longitude in degrees	Double	8	H+16
6	hgt	Master MSL Height in metres	Double	8	H+24
7	undulation	Undulation in metres	Float	4	H+32
8	datum id#	WGS84 (default) (refer to *Table 23, Reference Ellipsoid Constants* on *page 119*)	Enum	4	H+36
9	lat σ	Latitude Std in metres	Float	4	H+40
10	long σ	Longitude Std in metres	Float	4	H+44
11	hgt σ	Height Std in metres	Float	4	H+48
12	stn id	Receiver ID can be set using the DGPSTXID command, see *page 125*	Char[4]	4	H+52
13	Reserved		Float	4	H+56
14			Float	4	H+60
15	#SVs	Number of satellite vehicles tracked	Uchar	1	H+64
16	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+65
17	#obs	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+66
18	#multi	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+67
19	sol source	Solution source (see *Table 109, Solution Source* on *page 502*)	Hex	1	
20	Reserved		Uchar	1	H+69
21			Uchar	1	H+70
22			Uchar	1	H+71
23	xxxx	320bit CRC (ASCII and Binary only)	HEX	1	H+72
24	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

3.2.79 MATCHEDPOS
Matched RTK position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log represents positions that have been computed from time matched base and rover observations. There is no base extrapolation error on these positions because they are based on buffered measurements, that is, they lag real time by some amount depending on the latency of the data link. If the rover receiver has not been enabled to accept RTK differential data or is not actually receiving data leading to a valid solution, this is shown in fields #2 (*sol status*) and #3 (*pos type*).

This log provides the best accuracy in static operation. For lower latency in kinematic operation, see the RTKPOS (*page 649*) or BESTPOS (*page 406*) logs. The data in the logs changes only when a base observation (RTCM, RTCMV3, RTCA, CMRPLUS or CMR) changes.

A good message trigger for this log is ONCHANGED. Then, only positions related to unique base station messages are produced and the existence of this log indicates a successful link to the base.

> (i) Asynchronous logs, such as MATCHEDPOS, should only be logged ONCHANGED otherwise the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

Message ID: 96
Log Type: Asynch

Recommended Input:

> log matchedposa onchanged

ASCII Example:

```
#MATCHEDPOSA,COM1,0,63.0,FINESTEERING,1419,340034.000,00000040,
2F06,2724;SOL_COMPUTED,NARROW_INT,51.11635908660,-114.03833102484,
1063.8400,-16.2712,WGS84,0.0140,0.0075,0.0174,"AAAA",0.000,0.000,
12,12,12,12,0,01,0,33*FEAC3A3A
```

> (i) Measurement precision is different from the position computation precision. Measurement precision is a value that shows how accurately the actual code or carrier phase is measured by the GNSS receiver. Position precision is a value that shows the accuracy of the position computation made from the code and/or carrier phase measurements. The P-code L2 measurement precision is not as good as the C/A measurement precision because the NovAtel GNSS receiver is a civilian grade GPS device and does not have direct access to the decrypted military L2 P(Y) code. This means that NovAtel's semi-codeless P-code L2 measurements are noisier than the civilian band C/A code measurements. Refer to the OEM6 Installation and Operation Manual (OM-20000128) for the technical specification of the OEM6 card.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	MATCHED-POS header	Log header		H	0
2	sol status	Solution status (see *Table 85* on *page 408*)	Enum	4	H
3	pos type	Position type (see *Table 84* on *page 407*)	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 (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m) [a]	Float	4	H+32
8	datum id#	Datum ID number (see *Table 23, Reference Ellipsoid Constants* on *page 119*)	Enum	4	H+36
9	lat σ	Latitude standard deviation (m)	Float	4	H+40
10	lon σ	Longitude standard deviation (m)	Float	4	H+44
11	hgt σ	Height standard deviation (m)	Float	4	H+48
12	stn id	Base station ID	Char[4]	4	H+52
13	Reserved		Float	4	H+56
14			Float	4	H+60
15	#SVs	Number of satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+65
17	#ggL1	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		Hex	1	H+68
20	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	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.

3.2.80   MATCHEDSATS
Satellites used in MATCHEDPOS solution

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log lists the used and unused satellites for the corresponding MATCHEDPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

Message ID: **1176**
Log Type: **Asynch**

Recommended Input:

>   log matchedsats onchanged

Abbreviated ASCII Example:

```
<MATCHEDSATS COM1 0 60.5 FINESTEERING 1728 524924.000 00000000 B555
11487
<     24
<        GPS 3 GOOD 00000003
<        GPS 5 GOOD 00000003
...
<        GPS 23 GOOD 00000003
<        GPS 30 GOOD 00000003
<        GLONASS 1+1 GOOD 00000003
<        GLONASS 2-4 GOOD 00000003
...
<        GLONASS 21+4 GOOD 00000003 <
         BEIDOU 6 GOOD 00000003
<        BEIDOU 11 GOOD 00000003
...
<        BEIDOU 12 GOOD 00000003
<        BEIDOU 13 GOOD 00000003
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	MATCHEDSATS header	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	system	See *Table 110* on *page 508*	Enum	4	H+4
4	Satellite ID	Satellite identifier	Ulong	4	H+8
5	Status	Satellite status (*Table 90* on *page 413*)	Enum	4	H+12
6	Signal mask	See *Table 91* through *Table 94*	Hex	4	H+16
7	Next satellite offset = H + 4 + (#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	1	H+4+ (#sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.81 MATCHEDXYZ
Matched RTK Cartesian position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the receiver's matched position in ECEF coordinates. It represents positions that have been computed from time matched base and rover observations. There is no base station extrapolation error on these positions because they are based on buffered measurements, that is, they lag real time, by some amount, depending on the latency of the data link. If the rover receiver has not been enabled to accept RTK differential data or is not actually receiving data leading to a valid solution, this is reflected by the code shown in field #2 (solution status) and #3 (position type). See *Figure 11, The WGS84 ECEF Coordinate System* on *page 422* for a definition of the ECEF coordinates.

This log provides the best accuracy in static operation. For lower latency in kinematic operation, see the BESTXYZ or RTKXYZ logs on *pages 420* and *654* respectively. The data in the logs changes only when a base observation (RTCM, RTCMV3, RTCA or CMR) changes.

The time stamp in the header is the time of the matched observations that the computed position is based on and not the current time.

Message ID:	**242**
Log Type:	**Asynch**

Recommended Input:

> log matchedxyza onchanged

> (i) Asynchronous logs, such as MATCHEDXYZ, should only be logged ONCHANGED otherwise the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

ASCII Example:
```
#MATCHEDXYZA,COM1,0,62.5,FINESTEERING,1419,340035.000,00000040,
B8ED,2724;SOL_COMPUTED,NARROW_INT,-1634531.5703,-3664618.0321,
4942496.3280,0.0080,0.0159,0.0154,"AAAA",12,12,12,12,0,01,0,
33*E4B84015
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	MATCHEDXYZ header	Log header		H	0
2	P-sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X σ	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y σ	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z σ	Standard deviation of P-Z (m)	Float	4	H+40
10	stn ID	Base station ID	Char[4]	4	H+44
11	#SVs	Number of satellites tracked	Uchar	1	H+48
12	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+49
13	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+50
14	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+51
15	Reserved		Char	1	H+52
16	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+53
17	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+54
18	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+55
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+56
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.82   NAVIGATE
User navigation data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log reports the status of the waypoint navigation progress. It is used in conjunction with the SETNAV command, see *page 296*.
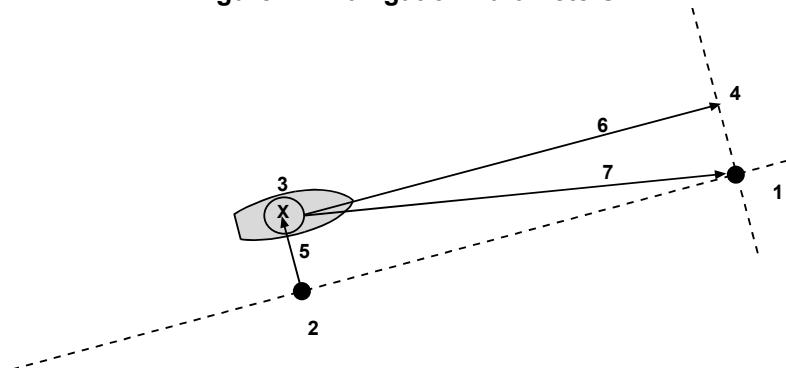
See the figure below for an illustration of navigation parameters.

> ▼ The SETNAV command must be enabled before valid data will be reported from this log.

Message ID: **161**
Log Type: **Sync**

Figure 12: Navigation Parameters

Reference	Description
1	TO lat-lon
2	X-Track perpendicular reference point
3	Current GPS position
4	A-Track perpendicular reference point
5	X-Track (cross track)
6	A-Track (along track)
7	Distance and bearing from 3 to 1

Table 118: Navigation Data Type

Navigation Data Type		Description
Binary	**ASCII**	
0	GOOD	Navigation is good
1	NOVELOCITY	Navigation has no velocity
2	BADNAV	Navigation calculation failed for an unknown reason
3	FROM_TO_SAME	"From" is too close to "To" for computation
4	TOO_CLOSE_TO_TO	Position is too close to "To" for computation
5	ANTIPODAL_WAYPTS	Waypoints are antipodal on surface

Recommended Input:

> log navigatea ontime 1

ASCII Example:

```
#NAVIGATEA,COM1,0,56.0,FINESTEERING,1337,399190.000,00000000,AECE,
1984;SOL_COMPUTED,PSRDIFF,SOL_COMPUTED,GOOD,9453.6278,303.066741,
133.7313,9577.9118,1338,349427.562*643CD4E2
```

> (i) Use the NAVIGATE log in conjunction with the SETNAV command to tell you where you currently are with relation to known To and From points. You can find a specific latitude, longitude or height knowing from where you started. For example, a backpacker could use these two commands to program a user supplied graphical display, on a digital GPS compass, to show their progress as they follow a defined route.

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	NAVIGATE header	Log header		H	0
2	sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+4
4	vel status	Velocity status, see *Table 85, Solution Status* on *page 408*	Enum	4	H+8
5	nav type	Navigation data type (see *Table 118, Navigation Data Type* on *page 545*)	Enum	4	H+12
6	distance	Straight line horizontal distance from current position to the destination waypoint, in metres (see *Figure 12, Navigation Parameters* on *page 545)*. This value is positive when approaching the waypoint and becomes negative on passing the waypoint	Double	8	H+16
7	bearing	Direction from the current position to the destination waypoint, in degrees, with respect to True North (or magnetic if corrected for magnetic variation by MAGVAR command)	Double	8	H+24
8	along track	Horizontal track distance from the current position to the closest point on the waypoint arrival perpendicular; expressed in metres. This value is positive when approaching the waypoint and becomes negative on passing the waypoint	Double	8	H+32

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
9	xtrack	The horizontal distance (perpendicular track error), from the vessel's present position to the closest point on the great circle line, that joins the FROM and TO waypoints. If a "track offset" has been entered in the SETNAV command, xtrack is the perpendicular error from the "offset track". Xtrack is expressed in metres. Positive values indicate the current position is right of the Track, while negative offset values indicate left	Double	8	H+40
10	eta week	Estimated GPS reference week number at time of arrival at the "TO" waypoint, along track arrival perpendicular based on current position and speed, in units of GPS reference weeks. If the receiving antenna is moving at a speed of less than 0.1 m/s ,in the direction of the destination, the value in this field is "9999"	Ulong	4	H+48
11	eta secs	Estimated GPS seconds into week at time of arrival at destination waypoint along track arrival perpendicular, based on current position and speed, in units of GPS seconds into the week. If the receiving antenna is moving at a speed of less than 0.1 m/s in the direction of the destination, the value in this field is "0.000"	Double	8	H+52
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+60
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.83 NMEA Standard Logs

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains NMEA logs.

GLMLA	**GLONASS ALMANAC DATA**
GPALM	**ALMANAC DATA**
GPGGA	**GLOBAL POSITION SYSTEM FIX DATA AND UNDULATION**
GPGGALONG	**GPS FIX DATA, EXTRA PRECISION AND UNDULATION**
GPGGARTK	**GPS FIX DATA**
GPGLL	**GEOGRAPHIC POSITION**
GPGRS	**GPS RANGE RESIDUALS FOR EACH SATELLITE**
GPGSA	**GPS DOP AN ACTIVE SATELLITES**
GPGST	**PSEUDORANGE MEASUREMENT NOISE STATISTICS**
GPGSV	**GPS SATELLITES IN VIEW**
GPHDT	**NMEA HEADING LOG (ALIGN)**
GPRMB	**NAVIGATION INFORMATION**
GPRMC	**GPS SPECIFIC INFORMATION**
GPVTG	**TRACK MADE GOOD AND GROUND SPEED**
GPZDA	**UTC TIME AND DATE**

The NMEA log structures follow format standards as adopted by the National Marine Electronics Association. The reference document used is "Standard For Interfacing Marine Electronic Devices NMEA 0183 Version 3.01". For further information, refer to the Standards and References section of our website www.novatel.com/support/. The following table contains excerpts from Table 6 of the NMEA Standard which defines the variables for the NMEA logs. The actual format for each parameter is indicated after the description.

> (i) See the GPGGA usage box that applies to all NMEA logs on *page 474*.

<table>
<tr><td rowspan="5">ⓘ</td><td>1.</td><td>Spaces may only be used in variable text fields.</td></tr>
<tr><td>2.</td><td>A negative sign "-" (HEX 2D) is the first character in a Field if the value is negative. The sign is omitted if the value is positive.</td></tr>
<tr><td>3.</td><td>All data fields are delimited by a comma (,).</td></tr>
<tr><td>4.</td><td>Null fields are indicated by no data between two commas (,,). Null fields indicate invalid data or no data available.</td></tr>
<tr><td>5.</td><td>The NMEA Standard requires that message length be limited to 82 characters.</td></tr>
</table>

Field Type	Symbol	Definition
Special Format Fields		
Status	A	Single character field: A = Yes, Data Valid, Warning Flag Clear V = No, Data Invalid, Warning Flag Set
Latitude	llll.ll	Fixed/Variable length field: degrees|minutes.decimal - 2 fixed digits of degrees, 2 fixed digits of mins and a <u>variable</u> number of digits for decimal-fraction of mins. Leading zeros always included for degrees and mins to maintain fixed length. The decimal point and associated decimal-fraction are optional if full resolution is not required
Longitude	yyyyy.yy	Fixed/Variable length field: degrees|minutes.decimal - 3 fixed digits of degrees, 2 fixed digits of mins and a <u>variable</u> number of digits for decimal-fraction of mins. Leading zeros always included for degrees and mins to maintain fixed length. The decimal point and associated decimal-fraction are optional if full resolution is not required
Time	hhmmss.ss	Fixed/Variable length field: hours|minutes|seconds.decimal - 2 fixed digits of hours, 2 fixed digits of mins, 2 fixed digits of seconds and <u>variable</u> number of digits for decimal-fraction of seconds. Leading zeros always included for hours, mins and seconds to maintain fixed length. The decimal point and associated decimal-fraction are optional if full resolution is not required.
Defined field		Some fields are specified to contain predefined constants, most often alpha characters. Such a field is indicated in this standard by the presence of one or more valid characters. Excluded from the list of allowable characters are the following which are used to indicate field types within this standard: "A", "a", "c", "hh", "hhmmss.ss", "llll.ll", "x", "yyyyy.yy"
Numeric Value Fields		
Variable numbers	x.x	Variable length integer or floating numeric field. Optional leading and trailing zeros. The decimal point and associated decimal-fraction are optional if full resolution is not required (example: 73.10 = 73.1 = 073.1 = 73)
Fixed HEX field	hh___	Fixed length HEX numbers only, MSB on the left
Information Fields		

Field Type	Symbol	Definition
Variable text	c--c	Variable length valid character field
Fixed alpha field	aa___	Fixed length field of uppercase or lowercase alpha characters
Fixed number field	xx___	Fixed length field of numeric characters
Fixed text field	cc___	Fixed length field of valid characters

3.2.84 NOVATELXOBS
NovAtel proprietary RTK correction

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

NOVATELX is a combined observation and reference station message used to transmit RTK corrections. This is a proprietary message and supports sending data for all systems.

Message ID: **1618**
Log Type: **Synch**

Recommended Input:

> log com2 novatelxobs ontime 1

To calculate the size of the NOVATELXOBS messages, use the following formula.

   Size = 168 + s * (6 + p * (54 + f*33))

where:

s = number of systems (maximum 8)

p = number of PRN per system (maximum 64)

f = number of signals data per PRN – 1 (maximum 10 signals possible)

Example size calculations:
- For 2 systems (GPS and GLONASS), 12 PRN per system, and 2 signals per satellite (L1CA, L2PY)
  Size = 168 + 2 * (6 + 12 * (54 + 33))
       = 2268 bits per second
       = 284 bytes + NovAtelXHeader (8 bytes)

- For 3 systems (GPS, BEIDOU and GLONASS), 12 PRN per system, and 2 signals per satellite (L1CA, L2PY)
  Size = 168 + 3 * (6 + 12 * (54 + 33))
       = 3318 bits per second
       = 415 bytes + NovAtelXHeader (8 bytes)

- For 3 systems (GPS, BEIDOU and GLONASS), 12 PRN per system, and 3 signals per satellite (L1CA, L2PY, L2C)
  Size = 168 + 3 * (6 + 12 * (54 + 2*33))
       = 4506 bits per second
       = 564 bytes + NovAtelXHeader (8 bytes)

3.2.85 NOVATELXREF
NovAtel proprietary reference station message for use in ALIGN

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

NOVATELXREF is a proprietary extended reference station message for use in ALIGN configurations only. This message enables the output of MASTERPOS, ROVERPOS and ALIGNBSLNENU logs on the rover.

Message ID: **1620**
Log Type: **Synch**

Recommended Input:

         log com2 novatelxref ontime 1

3.2.86 OMNIHPPOS
OmniSTAR HP/XP/G2 position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log outputs OmniSTAR Extra Performance (XP), High Performance (HP) or G2 position information.

> (i) In addition to a NovAtel receiver with L-Band capability, a subscription to an OmniSTAR service is required. Contact NovAtel for details, through our website at www.novatel.com through *Support*, or the *Customer Support* section in the OEM6 Family Installation and Operation User Manual (OM-20000128).

Message ID: **495**
Log Type: **Synch**

Recommended Input:

log omnihpposa ontime 1

ASCII Example:

```
#OMNIHPPOSA,COM1,0,67.5,FINESTEERING,1419,320435.000,00000000,808D,
2724;SOL_COMPUTED,OMNISTAR_HP,51.11635489609,-114.03819540112,
1063.8314,-16.2713,WGS84,0.1258,0.2135,0.2342,"1000",8.000,0.000,
13,10,10,10,0,00,0,03*E8510806
```

.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	OMNIHPPOS header	Log header		H	0
2	sol status	Solution status, see *Table 85* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84* on *page 407*	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 (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m) [a]	Float	4	H+32
8	datum id#	Datum ID number (see *Chapter 2, Table 23, Reference Ellipsoid Constants* on *page 119*)	Enum	4	H+36
9	lat σ	Latitude standard deviation (m)	Float	4	H+40
10	lon σ	Longitude standard deviation (m)	Float	4	H+44
11	hgt σ	Height standard deviation (m)	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

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
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 satellite vehicles used in solution	Uchar	1	H+65
17	#ggL1	Number of GPS plus GLONASS L1 used in solution	Uchar	1	H+66
18	#ggL1L2	Number of GPS plus GLONASS L1 and L2 used in solution	Uchar	1	H+67
19	Reserved		Hex	1	H+68
20	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+71
23	sig mask	Signals used mask - if 0, signals used in solution are unknown (see *Table 86* on *page 409*)	Hex	1	H+71
24	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
25	[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.

3.2.87 OMNIHPSATS
Satellites used in the OMNIHPPOS solution

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log lists the used and unused satellites for the corresponding OMNIHPPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

Message ID: **1197**
Log Type: **Synch**

Recommended Input:

> log omnihpsatsa ontime 1

ASCII Example:

```
#OMNIHPSATSA,COM2,0,85.0,FINESTEERING,1610,241351.000,00000000,2577
,6293;17,GPS,17,GOOD,00000003,GPS,2,GOOD,00000003,GPS,10,GOOD,00000
003,GPS,4,GOOD,00000003,GPS,29,GOOD,00000003,GPS,5,NOTUSED,00000000
,GPS,25,GOOD,00000003,GPS,9,GOOD,00000003,GPS,12,GOOD,00000003,GLON
ASS,15,NOTUSED,00000000,GLONASS,2-4, NOTUSED,00000000,GLONASS,18-3,
NOTUSED,00000000,GLONASS,9-2,NOTUSED, 00000000,GLONASS,1+1,NOTUSED,
00000000,GLONASS,17+4,NOTUSED,00000000,GLONASS,16-1,NOTUSED,
00000000,GLONASS,19+3,NOTUSED,00000000*F9EB4605
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	OMNIHPSATS header	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	system	See *Table 110, Satellite System* on *page 508*.	Enum	4	H+4
4	Satellite ID	Satellite identifier	Ulong	4	H+8
5	Status	Satellite status. See *Table 90* on *page 413*	Enum	4	H+16
6	Status mask	See *Table 91* on *page 414* to *Table 94* on *page 414*	Hex	4	H+20
7	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#entries x 16)
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.88 OMNIVIS
Omnistar satellite visibility list

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains OmniSTAR satellite and visibility information.

> (i) For local OmniSTAR beams, the satellite with the smallest local ellipsoid distance is the best one to use. For global beams, the satellite with the highest elevation is the best one.

Message ID:	**860**
Log Type:	**Synch**

Recommended Input:

    log omnivisa ontime 1

ASCII Example 1:

```
#OMNIVISA,COM1,0,60.5,FINESTEERING,1419,396070.000,00000020,0041,
2710;TRUE,8,
10,0,"MSVW_",0,0.000,1536782000,1200,C685,-1.16,-90.00,
11,0,"MSVC_",0,0.000,1534741000,1200,C685,8.28,-90.00,
12,0,"MSVE_",0,0.000,1530359000,1200,C685,22.97,-90.00,
8,0,"AMSAT",0,0.000,1535137500,1200,C685,34.87,31.09,
7,0,"EASAT",0,0.000,1535152500,1200,C685,91.01,-41.76,
3,0,"AFSAT",0,0.000,1535080000,1200,C685,110.73,-41.76,
4,0,"APSAT",0,0.000,1535137500,1200,2873,185.25,-40.66,
13,0,"OCSAT",0,0.000,1535185000,1200,2873,235.91,-18.57*B35C9CDF
```

ASCII Example 2:

```
#OMNIVISA,COM1,0,62.5,FINESTEERING,1419,334202.000,00000020,0041,
2710;FALSE,0*9E0F9078
```

> (i) Local Beams: When the value is negative, the user is inside the local beam footprint and a signal should be available. Beams with small positive values may be available but their availability is not guaranteed.
>
> Global Beams: Any beams above 0 degrees are visible, however the tracking may be marginal for elevations less than 10 degrees.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	OMNIVIS header	Log header		H	0
2	valid	Is the list of satellites valid? 0 = FALSE 1 = TRUE	Bool	4	H
3	#recs	Number of records to follow	Ulong	4	H+4
4	link ID	Satellite link ID	Uchar	1	H+8
5	app flag	Time of applicability flag: 0 = Valid Now 1 = Invalid 2 = Valid Until 3 = Valid After 4-7 = Reserved	Uchar	1	H+9
6	sat name	Satellite name	String [max 6]	variable[a]	H+10
7	app week	Time of applicability week	Ulong	4	H+16
8	app sec	Time of applicability (s into the week)	GPSec	4	H+20
9	freq	Satellite broadcast frequency (Hz)	Ulong	4	H+24
10	bit rate	Satellite data bit rate	Ushort	2	H+28
11	service id	Satellite service ID	Hex	2	H+30
12	ellip dist	Local ellipsoid distance parameter (kilometres)	Float	4	H+32
13	global elev	Global beam elevation (degrees)	Float	4	H+36
14	Next satellite offset = H+8+(#recs x 32)				
15	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+8+ (#recs x 32)
16	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

3.2.89   *PASSCOM, PASSXCOM, PASSAUX, PASSUSB, PASSETH1, PASSICOM, PASSNCOM*

Redirects data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The pass-through logging feature enables the receiver to redirect any ASCII or binary data, input at a specified port, to any specified receiver port. It allows the receiver to perform bi-directional communications with other devices such as a modem, terminal or another receiver. See also the INTERFACEMODE command on *page 184*.

There are many pass through logs: PASSCOM1, PASSCOM2, PASSCOM3, PASSCOM4, PASSCOM5, PASSCOM6, PASSXCOM1, PASSXCOM2, PASSXCOM3, PASSAUX, PASSETH1, PASSICOM1, PASSICOM2, PASSICOM3, PASSNCOM1, PASSNCOM2, PASSNCOM3 allow for redirection of data that is arriving at COM1, COM2, COM3, virtual COM1, virtual COM2 or AUX. PASSCOM4, PASSCOM5 and PASSCOM6 are only available on OEM638 receivers. The AUX port is available on some products. PASSUSB1, PASSUSB2, PASSUSB3 are only available on receivers supporting USB and can be used to redirect data from USB1, USB2 or USB3. PASSETH1 is only available on receivers supporting Ethernet and can be used to redirect data from ETH1.

A pass through log is initiated the same as any other log, that is, LOG [to-port] [data-type] [trigger]. However, pass-through can be more clearly specified as: LOG [to-port] [from-port-AB] [onchanged]. Now, the [from-port-AB] field designates the port which accepts data (that is, COM1, COM2, COM3, COM4, COM5, COM6, AUX, USB1, USB2 or USB3) as well as the format in which the data is logged by the [to-port] (A for ASCII or B for Binary).

> ⓘ To pass through data arriving on all ports, use the PASSTHROUGH log (*page 563*).

When the [from-port-AB] field is suffixed with an [A], all data received by that port is redirected to the [to-port] in ASCII format and logs according to standard NovAtel ASCII format. Therefore, all incoming ASCII data is redirected and output as ASCII data. However, any binary data received is converted to a form of ASCII hexadecimal before it is logged.

When the [from-port-AB] field is suffixed with a [B], all data received by that port is redirected to the [to-port] exactly as it is received. The log header and time tag adhere to standard NovAtel Binary format followed by the pass through data as it was received (ASCII or binary).

Pass through logs are best utilized by setting the [trigger] field as onchanged or onnew.

If the data being injected is ASCII, then the data is grouped together with the following rules:

- blocks of 80 characters
- any block of characters ending in a <CR>
- any block of characters ending in a <LF>
- any block remaining in the receiver code when a timeout occurs (100 ms)

If the data being injected is binary or the port INTERFACEMODE mode is set to GENERIC, then the data is grouped as follows:

- blocks of 80 bytes
- any block remaining in the receiver code when a timeout occurs (100 ms)

If a binary value is encountered in an ASCII output, then the byte is output as a hexadecimal byte preceded by a backslash and an x. For example 0A is output as \x0A. An actual '\' in the data is output as \\. The output counts as one pass through byte although it is four characters.

The first character of each pass-through record is time tagged in GPS reference weeks and seconds.

PASSCOM1 Message ID: 233

PASSCOM2 Message ID: 234

PASSCOM3 Message ID: 235

PASSCOM4 Message ID: 1384

PASSCOM5 Message ID: 1576

PASSCOM6 Message ID: 1577

PASSXCOM1 Message ID: 405

PASSXCOM2 Message ID: 406

PASSXCOM3 Message ID: 795

PASSUSB1 Message ID: 607

PASSUSB2 Message ID: 608

PASSUSB3 Message ID: 609

PASSAUX Message ID: 690

PASSETH1 Message ID: 1209

PASSICOM1 Message ID: 1250

PASSICOM2 Message ID: 1251

PASSICOM3 Message ID: 1252

PASSNCOM1 Message ID: 1253

PASSNCOM2 Message ID: 1254

PASSNCOM3 Message ID: 1255

PASSCOM7 Message ID: 1701

PASSCOM8 Message ID: 1702

PASSCOM9 Message ID: 1703

PASSCOM10 Message ID: 1704

Log Type: Asynch

Recommended Input:

```
log passcom1a onchanged
```

> ⓘ Asynchronous logs should only be logged ONCHANGED otherwise the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

ASCII Example 1:

```
#PASSCOM2A,COM1,0,59.5,FINESTEERING,1337,400920.135,00000000,2B46,
1984;80,#BESTPOSA,COM3,0,80.0,FINESTEERING,1337,400920.000,
00000000,4CA6,1899;SOL_COMPUT*F9DFAB46
#PASSCOM2A,COM1,0,64.0,FINESTEERING,1337,400920.201,00000000,2B46,
1984;80,ED,SINGLE,51.11636326036,-114.03824210485,1062.6015,
-16.2713,WGS84,1.8963,1.0674*807FD3CA
#PASSCOM2A,COM1,0,53.5,FINESTEERING,1337,400920.856,00000000,2B46,
1984;49,,2.2862,"",0.000,0.000,9,9,0,0,0,0,0,0*20B24878\X0D\X0A*3EE
F4220
#PASSCOM1A,COM1,0,53.5,FINESTEERING,1337,400922.463,00000000,13FF,
1984;17,UNLOG PASSCOM2A\X0D\X0A*EF8D2508
```

ASCII Example 2:
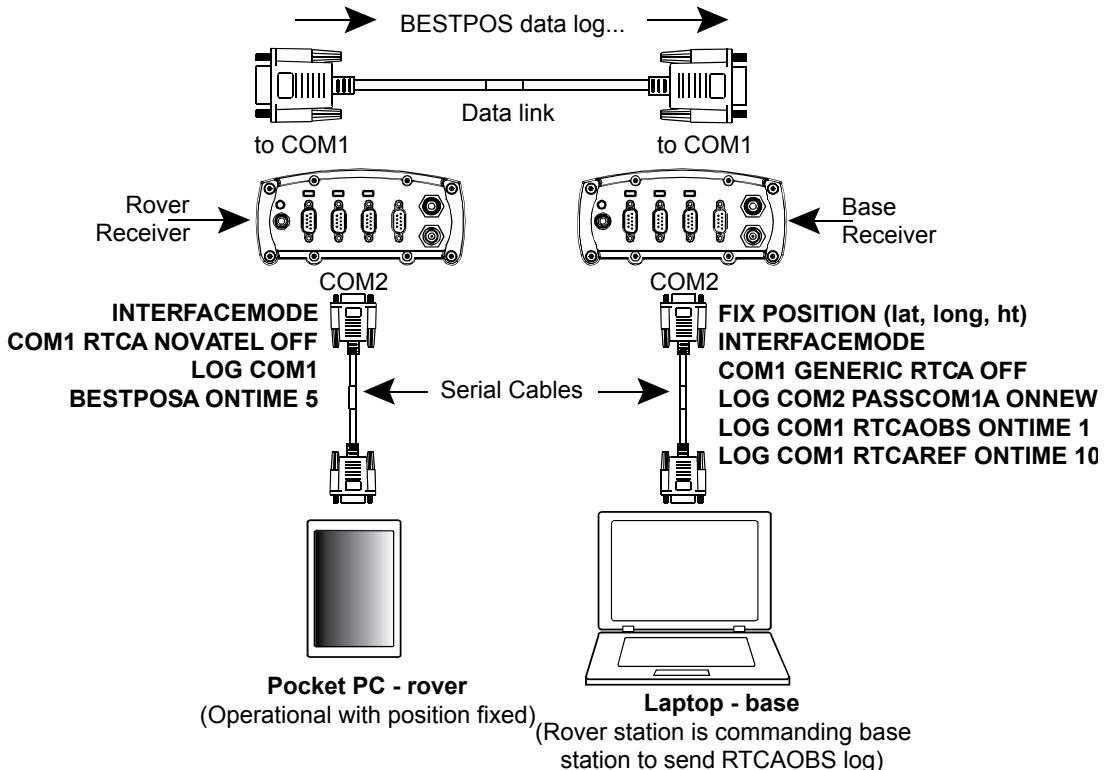
```
#PASSCOM2A,COM1,0,53.0,FINESTEERING,1337,400040.151,00000000,2B46,
1984;80,\X99A\X10\X04\X07YN &\XC6\XEA\XF10\X00\X01\XDE\X00\X00\
X10\XFE\XBF\XFE1\XFE\X9C\XF4\X03\XE2\XEF\X9F\X1F\XF3\XFF\XD6\XFF\
XC3_A~Z \XAA\XFE\XBF\XF9\XD3\XF8\XD4\XF4-\XE8KHO\XE2\X00>\XE0QOC>\
XC3\X9C\X11\XFF\X7F\XF4\XA1\XF3T\XF4'\XF4XVO\XE6\X00\X9D*DCD2E989
```

In the example, note that '~' is a printable character.

> ⓘ For example, you could connect two OEM6 family receivers together via their COM1 ports such as in the *Figure 13* (a rover station to base station scenario). If the rover station is logging BESTPOSA data to the base station, it is possible to use the pass through logs to pass through the received BESTPOSA data to a disk file (let's call it diskfile.log) at the base station host PC hard disk.

Figure 13: Pass Through Log Data

BESTPOS data log...

Data link

to COM1   to COM1

Rover
Receiver

Base
Receiver

COM2   COM2

**INTERFACEMODE
COM1 RTCA NOVATEL OFF
LOG COM1
BESTPOSA ONTIME 5**

Serial Cables

**FIX POSITION (lat, long, ht)
INTERFACEMODE
COM1 GENERIC RTCA OFF
LOG COM2 PASSCOM1A ONNEW
LOG COM1 RTCAOBS ONTIME 1
LOG COM1 RTCAREF ONTIME 10**

Pocket PC - rover
(Operational with position fixed)

Laptop - base
(Rover station is commanding base
station to send RTCAOBS log)

Under default conditions, the two receivers "chatter" back and forth with the Invalid
Command Option message (due to the command interpreter in each receiver not
recognizing the command prompts of the other receiver). The chattering in turn causes the
accepting receiver to transmit new pass through logs with the response data from the other
receiver. To avoid the chattering problem, use the INTERFACEMODE command on the
accepting port to disable error reporting from the receiving port command interpreter.

If the accepting port's error reporting is disabled by INTERFACEMODE, the BESTPOSA
data record passes through and creates two records.

The reason that two records are logged from the accepting receiver is the first record was
initiated by receipt of the BESTPOSA first terminator <CR>. The second record followed in
response to the BESTPOSA second terminator <LF>.

Note the time interval between the first character received and the terminating <LF> can be
calculated by differencing the two GPS REFERENCE TIME tags. This pass through feature
is useful for time tagging the arrival of external messages. These messages can be any user
related data. When using this feature for tagging external events, it is recommended that the
rover receiver be disabled from interpreting commands so the receiver does not respond to
the messages, using the INTERFACEMODE command, see *page 184*.

If the BESTPOSB binary log data is input to the accepting port (log com2 passcom1a onchanged), the BESTPOSB binary data at the accepting port is converted to a variation of ASCII hexadecimal before it is passed through to COM2 port for logging.

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	PASSCOM header	Log header		H	0
2	#bytes	Number of bytes to follow	Ulong	4	H
3	data	Message data	Char [80]	80	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+(#bytes)
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.90 PASSTHROUGH
Redirected data from all ports

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log outputs pass through data from all receiver ports. The behavior is the same as the port specific pass though logs described in *Section 3.2.89* on *page 558*.

Message ID: 1342

Log Type: Asynch

Recommended Input:

log passthrougha onchanged

ASCII Example:

```
#PASSTHROUGHA,COM1,0,73.0,FINESTEERING,1625,165965.067,00040008,5FA
3,39275;USB1,80,I\XD3\X00\X87>\XB0\X00'\X91\XB3"\XA0D?\XAA\XB2\X00\
X07OP\X18@\X05\XE9\XD4\X08\XE7\X03\X7F\XFD\X18{\X80W\XFF\XF2N_CY\X1
1\X80\X0BC\XDC\X01@\X00\XDFR\XB1`\X873\XFF\X81]\X7F\XE3\XFF\XEA\X83
V\X08M\XD8?\XFCR\XF7\X01\X18\X00\X17\X1D2\XD1\XD1B\X00*5CB8BD9A
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PASSTHROUGH header	Log header		H	0
2	Port	See *Table 57, COM Port Identifiers* on *page 284*	Enum	4	H
3	#bytes	Number of bytes to follow	Ulong	4	H+4
4	data	Message data	Char[80]	80	H+8
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+8+#bytes
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.91 PDPPOS
PDP filter position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The PDPPOS log contains the receiver position computed by the receiver with the PDP filter enabled. See also the `PDPFILTER` command on *page 227*.

Message ID: **469**

Log Type: **Synch**

Recommended Input:

> log pdpposa ontime 1

ASCII Example:

```
#PDPPOSA,COM1,0,75.5,FINESTEERING,1431,494991.000,00040000,A210,
35548; SOL_COMPUTED,SINGLE,51.11635010310,-114.03832575772,
1065.5019,-16.9000,WGS84,4.7976,2.0897,5.3062,"",0.000,0.000,8,8,
0,0,0,0,0,0*3CBFA646
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PDPPOS header	Log header		H	0
2	sol status	Solution status (refer to *Table 85, Solution Status* on *page 408*)	Enum	4	H
3	pos type	Position type (refer to *Table 84, Position or Velocity Type* on *page 407*)	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 (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m) [a]	Float	4	H+32
8	datum id#	Datum ID number (refer to *Table 24, Datum Transformation Parameters* on *page 120*)	Enum	4	H+36
9	lat σ	Latitude standard deviation (m)	Float	4	H+40
10	lon σ	Longitude standard deviation (m)	Float	4	H+44
11	hgt σ	Height standard deviation (m)	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	#sats	Number of satellites tracked	Uchar	1	H+64
16	#sats soln	Number of satellites in the solution	Uchar	1	H+65

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
17	Reserved		Uchar	1	H+66
18			Uchar	1	H+67
19			Hex	1	H+68
20	ext sol stat	Extended solution status (see *Table 88* on *page 410*)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	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.

3.2.92   PDPSATS
Satellites used in PDPPOS solution

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log lists the used and unused satellites for the corresponding PDPPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

Message ID: **1234**
Log Type: **Synch**

Recommended Input:

>   log pdpsatsa ontime 1

Abbreviated ASCII Example:
```
<PDPSATS COM1 0 80.0 FINESTEERING 1690 603073.000 00000008 BE33 43488
<      21
<          GPS 11 GOOD 00000001
<          GPS 27 GOOD 00000001
...
<          GPS 1 GOOD 00000001
<          GPS 7 GOOD 00000001
<          SBAS 133 NOTUSED 00000000
<          SBAS 138 NOTUSED 00000000
<          SBAS 135 NOTUSED 00000000
<          GLONASS 10-7 GOOD 00000001
<          GLONASS 21+4 GOOD 00000001
...
<          GLONASS 12-1 GOOD 00000001
<          GLONASS 11 GOOD 00000001
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PDPSATS header	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	system	See *Table 110* on *page 508*	Enum	4	H+4
4	Satellite ID	Satellite identifier	Ulong	4	H+8
5	Status	Satellite status (see *Table 90* on *page 413*)	Enum	4	H+12
6	Status mask	See *Table 91* through *Table 94*	Hex	4	H+16
7	Next satellite offset = H+4+(#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.93 PDPVEL
PDP filter velocity

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The PDPVEL log contains the pseudorange velocity computed by the receiver with the PDP filter enabled. See also the `PDPFILTER` command on *page 227*.

Message ID: **470**
Log Type: **Synch**

Recommended Input:

  log pdpvela ontime 1

ASCII Example:
```
#PDPVELA,COM1,0,75.0,FINESTEERING,1430,505990.000,00000000,B886,
2859;SOL_COMPUTED,SINGLE,0.150,0.000,27.4126,179.424617,-0.5521,
0.0*7746B0FE
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PDPVEL header	Log header		H	0
2	sol status	Solution status (refer to *Table 85, Solution Status* on *page 408*)	Enum	4	H
3	vel type	Velocity type (refer to *Table 84, Position or Velocity Type* on *page 407*)	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 in seconds	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	height	Height in metres 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)	Ulong	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.94 PDPXYZ
PDP filter Cartesian position and velocity

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The PDPXYZ log contains the Cartesian position in X, Y and Z coordinates as computed by the receiver with the PDP filter enabled. See also the PDPFILTER command (*page 227*).

Message ID: **471**
Log Type: **Synch**

Recommended Input:

>   log pdpxyza ontime 1

ASCII Example:

```
#PDPXYZA,COM1,0,75.5,FINESTEERING,1431,494991.000,00040000,33CE,
35548;SOL_COMPUTED,SINGLE,-1634531.8128,-3664619.4862,4942496.5025,
2.9036,6.1657,3.0153,SOL_COMPUTED,SINGLE,-2.5588E-308,-3.1719E-308,
3.9151E-308,0.0100,0.0100,0.0100,"",0.150,0.000,0.000,8,8,0,0,0,0,
0,0*A20DBD4F
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PDPXYZ header	Log header		H	0
2	P-sol status	Solution status (refer to *Table 85, Solution Status* on *page 408*)	Enum	4	H
3	pos type	Position type (refer to *Table 84, Position or Velocity Type* on *page 407*)	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X σ	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y σ	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z σ	Standard deviation of P-Z (m)	Float	4	H+40
10	V-sol status	Solution status (refer to *Table 85, Solution Status* on *page 408*)	Enum	4	H+44
11	vel type	Velocity type (refer to *Table 84, Position or Velocity Type* on *page 407*)	Enum	4	H+48
12	V-X	Velocity vector along X-axis (m)	Double	8	H+52
13	V-Y	Velocity vector along Y-axis (m)	Double	8	H+60
14	V-Z	Velocity vector along Z-axis (m)	Double	8	H+68
15	V-X σ	Standard deviation of V-X (m)	Float	4	H+76
16	V-Y σ	Standard deviation of V-Y (m)	Float	4	H+80

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
17	V-Z σ	Standard deviation of V-Z (m)	Float	4	H+84
18	stn ID	Base station ID	Char[4]	4	H+88
19	V-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+92
20	diff_age	Differential age in seconds	Float	4	H+96
21	sol_age	Solution age in seconds	Float	4	H+100
22	#sats	Number of satellite vehicles tracked	Uchar	1	H+104
23	#sats soln	Number of satellite vehicles used in solution	Uchar	1	H+105
24			Uchar	1	H+106
25	Reserved		Uchar	1	H+107
26			Uchar	1	H+108
27	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+109
28	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+110
29	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+111
30	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+112
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.95   PORTSTATS
Port statistics

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log conveys various status parameters of the receiver's communication ports. The receiver maintains a running count of a variety of status indicators of the data link. This log outputs a report of those indicators.

Message ID: **72**
Log Type: **Polled**

Recommended Input:

         log portstatsa once

ASCII example:

```
#PORTSTATSA,COM1,0,89.0,FINESTEERING,1609,500694.175,00000000,A872,
6259;15,COM1,313,3477,313,0,755,0,0,0,0,COM2,0,6,0,0,6,0,0,0,0,COM3
,0,6,0,0,6,0,0,0,0,USB1,0,0,0,0,0,0,0,0,0,USB2,0,0,0,0,0,0,0,0,0,US
B3,0,0,0,0,0,0,0,0,0,XCOM1,0,7,0,0,0,0,0,0,0,XCOM2,0,7,0,0,0,0,0,0,
0,XCOM3,0,7,0,0,0,0,0,0,0,ICOM1,0,0,0,0,0,0,0,0,0,ICOM2,0,0,0,0,0,0
,0,0,0,ICOM3,0,0,0,0,0,0,0,0,0,NCOM1,0,0,0,0,0,0,0,0,0,NCOM2,0,0,0,
0,0,0,0,0,0,NCOM3,0,0,0,0,0,0,0,0,0*8F33F6EF
```

> (i) Parity and framing errors occur for COM ports if poor transmission lines are encountered or if there is an incompatibility in the data protocol. If errors occur, you may need to confirm the bit rate, number of data bits, number of stop bits and parity of both the transmit and receiving ends. Characters may be dropped when the CPU is overloaded.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PORTSTATS header	Log header		H	0
2	#port	Number of ports with information to follow	Long	4	H
3	port	Refer to *Table 28, Communications Port Identifiers* on *page 141*	Enum	4	H+4
4	rx chars	Total number of characters received through this port	Ulong	4	H+8
5	tx chars	Total number of characters transmitted through this port	Ulong	4	H+12
6	acc rx chars	Total number of accepted characters received through this port	Ulong	4	H+16
7	dropped rx chars	Number of software overruns in receive	Ulong	4	H+20
8	interrupts	Number of interrupts on this port	Ulong	4	H+24

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
9	breaks	Number of breaks (only for serial ports)	Ulong	4	H+28
10	par err	Number of parity errors (only for serial ports)	Ulong	4	H+32
11	frame err	Number of framing errors (only for serial ports)	Ulong	4	H+36
12	rx overruns	Number of hardware overruns in receive	Ulong	4	H+40
13	Next port offset = H+4+(#port x 40)				
14	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#port x 40)
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.96 PPPPOS
PPP filter position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the position solution computed by the PPP filter.

Message ID: **1538**
Log Type: **Synch**

Recommended Input:

        log pppposa ontime 1

ASCII Example:

```
#PPPPOSA,COM1,0,80.0,FINESTEERING,
1735,345300.000,00000000,6F47,44027;
SOL_COMPUTED,PPP,51.11635350286,-114.03819287079,1064.5365,-
16.9000,WGS84,0.0375,0.0460,0.0603,"0",4.000,0.000,12,12,12,12,0,00
,00,03*EF17D668
```

Table 119: Position Type

ASCII	Binary	Description
NONE	0	No solution
PPP_CONVERGING	68	Converging PPP solution
PPP	69	Converged PPP solution

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PPPPOS header	Log header		H	0
2	Status	Solution status (see *Table 85, Solution Status* in this manual*)*	Enum	4	H
3	Type	Position type (see *Table 84, Position or Velocity Type* in this manual)	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 (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m)[a]	Float	4	H+32
8	datum id#	Datum ID number (see *Table 23, Reference Ellipsoid Constants* in this manual)	Enum	4	H+36
9	lat σ	Latitude standard deviation (m)	Float	4	H+40
10	lon σ	Longitude standard deviation (m)	Float	4	H+44
11	hgt σ	Height standard deviation (m)	Float	4	H+48
12	stn id	Base station ID	Char[4]	4	H+52

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
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 satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites vehicles used in solution	Uchar	1	H+65
17	#ggL1	Number of GPS plus GLONASS plus BDS L1/B1 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		Hex	1	H+68
20	ext sol stat	Extended solution status (see *Table 110, Satellite System* in this manual)	Hex	1	H+69
21	Reserved		Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* or *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* in this manual)	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	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.

3.2.97   PPPSATS
Satellites used in the PPPPOS solution

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log lists the used and unused satellites for the corresponding PPPPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

Message ID: **1541**
Log Type: **Synch**

Recommended Input:

        log pppsatsa ontime 1

Abbreviated ASCII Example:

```
<PPPSATS COM1 0 80.0 FINESTEERING 1735 345300.000 00000000 CE3F 44027
<     12
<          GPS 3 GOOD 00000003
<          GPS 5 GOOD 00000003
<          GPS 6 GOOD 00000003
<          GPS 7 GOOD 00000003
<          GPS 8 GOOD 00000003
<          GPS 10 GOOD 00000003
<          GPS 13 GOOD 00000003
<          GPS 16 GOOD 00000003
<          GPS 19 GOOD 00000003
<          GPS 23 GOOD 00000003
<          GPS 26 GOOD 00000003
<          GPS 28 GOOD 00000003
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PPPSATS header	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	System	Satellite system (see *Table 110, Satellite System* in this manual)	Enum	4	H+4
4	Satellite ID	See the description of "Satellite ID" in the BESTSATS field table	Ulong	4	H+8
5	Status	Satellite status (see *Table 90, Observation Statuses* in this manual)	Enum	4	H+12

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
6	Signal Mask	Signals used in the solution (see *Table 91, BESTSATS GPS Signal Mask*, *Table 92, BESTSATS GLONASS Signal Mask*, *Table 93, BESTSATS Galileo Signal Mask* and *Table 94, BESTSATS BeiDou Signal Mask* in this manual)	Hex	4	H+16
7	Next satellite offset = H + 4 + (#entries x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#entries x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.98   PROFILEINFO
Profile information in NVM

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log outputs a list of Profiles in the system. Refer also to the PROFILE command (*page 244*).

> (i)   A list may consist of a maximum of 20 profiles.

Message ID: 1412
Type: Polled

Recommended Input:

> log profileinfoa onchanged

ASCII Examples:

```
#PROFILEINFOA,COM1,0,84.0,UNKNOWN,0,17539.339,004C0020,AE3A,10526;
"BASE",0,2,
"LOG VERSION",
"SERIALCONFIG COM2 230400"*0AD5CDA5
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	PROFILEINFO header		-	H	0
2	Name	Profile Name	String [Max 20]	variable[a]	H
3	Status Word	Refer to *Table 120, Status Word* on *page 577*	Ulong	4	variable
4	# of Commands	# of Commands	Ulong	4	variable
5	Command #1	Profile command	String [Max 150]	variable[a]	variable
6	Next port offset = variable				
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	variable
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

ⓘ If the input command has an optional port input, COM1 is assumed as the input.

For example, if <Profile Createelement Testing "log versiona"> is entered, defaults as "Log COM1 Versiona".

If you want to log version on COM2, then <Profile Createelement Testing "Log Com2 versiona"> command has to be entered.

Table 120: Status Word

Bit #	MASK	Description
0	0x00000001	Activate Flag (0 – Deactivate (default), 1 –Activate)
1 – 3	0x0000000E	Reserved
4	0x00000010	Command 1 validation Flag (0 – Valid (default), 1 – Invalid)
5	0x00000020	Command 2 validation Flag
6	0x00000040	Command 3 validation Flag
7	0x00000080	Command 4 validation Flag
8	0x00000100	Command 5 validation Flag
9	0x00000200	Command 6 validation Flag
10	0x00000400	Command 7 validation Flag
11	0x00000800	Command 8 validation Flag
12	0x00001000	Command 9 validation Flag
13	0x00002000	Command 10 validation Flag
14	0x00004000	Command 11 validation Flag
15	0x00008000	Command 12 validation Flag
16	0x00010000	Command 13 validation Flag
17	0x00020000	Command 14 validation Flag
18	0x00040000	Command 15 validation Flag
19	0x00080000	Command 16 validation Flag
20	0x00100000	Command 17 validation Flag
21	0x00200000	Command 18 validation Flag
22	0x00400000	Command 19 validation Flag
23	0x00800000	Command 20 validation Flag
24 - 31	0xFF000000	Reserved

3.2.99 PSRDOP
Pseudorange DOP

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The dilution of precision data is calculated using the geometry of only those satellites currently being tracked and used in the position solution by the receiver. This log is updated once every 60 seconds. Therefore, the total number of data fields output by the log is variable and depends on the number of SVs being tracked.

> 1. If a satellite is locked out using the LOCKOUT command, it will still show in the PRN list but it will be significantly deweighted in the DOP calculation.
> 2. The vertical dilution of precision can be calculated by:
> $vdop = \sqrt{pdop^2 - hdop^2}$
> 3. If the DOP is not yet calculated, a default value of 9999.0 is displayed.

Message ID: **174**
Log Type: **Asynch**

Recommended Input:

> log psrdopa onchanged

ASCII Example:

```
#PSRDOPA,COM1,0,56.5,FINESTEERING,1337,403100.000,00000000,768F,
1984;1.9695,1.7613,1.0630,1.3808,0.8812,5.0,10,14,22,25,1,24,11,5,
20,30,7*106DE10A
```

> When operating in differential mode, you require at least four common satellites at the base and rover. The number of common satellites being tracked at large distances is less than at short distances. This is important because the accuracy of GPS and DGPS positions depend a great deal on how many satellites are being used in the solution (redundancy) and the geometry of the satellites being used (DOP). DOP stands for Dilution Of Precision and refers to the geometry of the satellites. A good DOP occurs when the satellites being tracked and used are evenly distributed throughout the sky. A bad DOP occurs when the satellites being tracked and used are not evenly distributed throughout the sky or grouped together in one part of the sky.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PSRDOP header	Log header		H	0
2	gdop	Geometric dilution of precision - assumes 3D position and receiver clock offset (all 4 parameters) are unknown	Float	4	H

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
3	pdop	Position dilution of precision - assumes 3D position is unknown and receiver clock offset is known	Float	4	H+4
4	hdop	Horizontal dilution of precision.	Float	4	H+8
5	htdop	Horizontal position and time dilution of precision.	Float	4	H+12
6	tdop	Time dilution of precision - assumes 3D position is known and only the receiver clock offset is unknown	Float	4	H+16
7	cutoff	GPS elevation cut-off angle	Float	4	H+20
8	#PRN	Number of satellites PRNs to follow	Long	4	H+24
9	PRN	PRN of SV PRN tracking, null field until position solution available	Ulong	4	H+28
10...	Next PRN offset = H+28+(#prn x 4)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+28+ (#prn x 4)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.100 PSRDOP2
Pseudorange DOP

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log is similar to the PSRDOP log but contains the per system Time Dilution of Precision (TDOP).

Message ID: **1163**
Log Type: **Asynch**

Recommended Input:

> log psrdop2a onchanged

ASCII Example:
```
#PSRDOP2A,COM1,0,89.5,FINESTEERING,1613,164820.000,00000008,0802,
39031;1.6740,1.3010,0.6900,1.1030,2,GPS,0.6890,GLONASS,
0.7980*5DD123D0.
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PSRDOP2 header	Log header		H	0
2	GDOP	Geometric dilution of precision - assumes 3D position and receiver clock offset (all 4 parameters) are unknown	Float	4	H
3	PDOP	Position dilution of precision - assumes 3D position is unknown and receiver clock offset is known	Float	4	H+4
4	HDOP	Horizontal dilution of precision	Float	4	H+8
5	VDOP	Vertical dilution of precision	Float	4	H+12
6	#systems	Number of systems	Ulong	4	H+16
7	system	See *Table 64, System Used for Timing* on *page 308*	Enum	4	H+20
8	TDOP	Time dilution of precision	Float	4	H+24
9	Next satellite offset = H+20+(#systems x 8)				
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+20+ (#systems x 8)
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.101 PSRPOS
Pseudorange position

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the position (in metres) computed by the receiver, along with three status flags. In addition, it reports other status indicators, including differential age, which is useful in predicting anomalous behavior brought about by outages in differential corrections.

Message ID: 47

Log Type: Synch

Recommended Input:

    log psrposa ontime 1

ASCII Example:

```
#PSRPOSA,COM1,0,58.5,FINESTEERING,1419,340037.000,00000040,6326,
2724;SOL_COMPUTED,SINGLE,51.11636177893,-114.03832396506,
1062.5470,-16.2712,WGS84,1.8532,1.4199,3.3168,"",0.000,0.000,12,
12,0,0,0,06,0,33*D200A78C
```

> ⓘ There are DGPS use cases in which the base receiver is not maintained or controlled by the positioning user. For example, the US Coast Guard operates a differential correction service which broadcasts GPS differential corrections over marine radio beacons. As a user, all you need is a marine beacon receiver and a GNSS receiver to achieve positioning accuracy of less than 1 metre. In this case, the Coast Guard owns and operates the base receiver at known coordinates. Other examples of users appearing to use only one GNSS receiver include FM radio station correction services, privately owned radio transmitters and corrections carried by communication satellites. Some of the radio receivers have built-in GNSS receivers and combined antennas, so they even appear to look as one self contained unit.
>
> The major factors degrading GPS signals which can be removed or reduced with differential methods are the atmosphere, ionosphere, satellite orbit errors, and satellite clock errors. Some errors which are not removed include receiver noise and multipath.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PSRPOS header	Log header		H	0
2	sol status	Solution status (see *Table 85, Solution Status* on *page 408*)	Enum	4	H
3	pos type	Position type (see *Table 84, Position or Velocity Type* on *page 407*)	Enum	4	H+4

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
4	lat	Latitude (degrees)	Double	8	H+8
5	lon	Longitude (degrees)	Double	8	H+16
6	hgt	Height above mean sea level (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m) [a]	Float	4	H+32
8	datum id#	Datum ID number (see *Table 23, Reference Ellipsoid Constants* on *page 119*)	Enum	4	H+36
9	lat σ	Latitude standard deviation (m)	Float	4	H+40
10	lon σ	Longitude standard deviation (m)	Float	4	H+44
11	hgt σ	Height standard deviation (m)	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 satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+65
17	Reserved		Uchar	1	H+66
18			Uchar	1	H+67
19			Hex	1	H+68
20	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	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.

3.2.102 PSRSATS
Satellites used in PSRPOS solution

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log lists the used and unused satellites for the corresponding PSRPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

Message ID: **1162**
Log Type: **Synch**

Recommended Input:

        log psrsats ontime 1

Abbreviated ASCII Example:

```
<PSRSATS COM1 0 80.0 FINESTEERING 1729 154910.000 00004000 FEA4 11465
<     20
<          GPS 31 GOOD 00000003
<          GPS 14 GOOD 00000003
<          GPS 22 GOOD 00000003
<          GPS 11 GOOD 00000003
<          GPS 1 GOOD 00000003
<          GPS 32 GOOD 00000003
<          GPS 18 GOOD 00000003
<          GPS 24 GOOD 00000003
<          GPS 19 GOOD 00000003
<          GLONASS 24+2 GOOD 00000003
<          GLONASS 10-7 GOOD 00000003
<          GLONASS 9-2 GOOD 00000003
<          GLONASS 2-4 GOOD 00000003
<          GLONASS 1+1 GOOD 00000003
<          GLONASS 11 GOOD 00000003
<          GLONASS 17+4 GOOD 00000003
<          GLONASS 18-3 GOOD 00000003
<          GALILEO 12 LOCKEDOUT 00000000
<          GALILEO 11 LOCKEDOUT 00000000
<          BEIDOU 8 GOOD 00000003
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PSRSATS header	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	system	See *Table 110* on *page 508*	Enum	4	H+4
4	Satellite ID	Satellite identifier	Ulong	4	H+8
5	Status	Satellite status (see *Table 90* on *page 413*)	Enum	4	H+12
6	Signal mask	See *Table 91* through *Table 94*	Hex	4	H+16
7	Next satellite offset = H+4+(#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.103 PSRTIME
Time offsets from the pseudorange filter

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the instantaneous receiver clock offsets calculated in the pseudorange filter for each GNSS used in the solution.

Message ID: **881**
Log Type: **Synch**

Recommended Input:

     log psrtimea ontime 1

ASCII Example:

```
#PSRTIMEA,COM1,0,62.5,FINESTEERING,1423,231836.000,00000000,462F,
35520;
2,
GPS,-1.2631E-09,7.1562E-09,
GLONASS,-7.0099E-07,2.4243E-08*40AA2AF1
```

> (i) Uses for this log include the following:
> 1. Estimating the difference between GPS and GLONASS satellite system times.
> 2. Estimating the difference between UTC and GLONASS system time.

.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PSRTIME header	Log header		H	0
2	#recs	Number of records to follow	Ulong	4	H
3	system	Navigation System 0 = GPS 1 = GLONASS (refer to *Table 64, System Used for Timing* on *page 308* for additional systems)	Enum	4	H+4
4	offset	GNSS time offset from the pseudorange filter (s)	Double	8	H+8
5	offset stdv	Time offset standard deviation (s)	Double	8	H+12
6	Next binary offset = H+4+(#recs x 20)				
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#recs x 20)
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.104 PSRVEL
Pseudorange velocity

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

In the PSRVEL log the actual speed and direction of the receiver antenna over ground is provided. 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. See also the table footnote for velocity logs on *page 362*.

The velocity in the PSRVEL log is determined by the pseudorange filter. Velocities from the pseudorange filter are calculated from the Doppler.

The velocity status indicates varying degrees of velocity quality. To ensure healthy velocity, the velocity sol-status must also be checked. If the sol-status is non-zero, the velocity is likely invalid. It should be noted that the receiver does not determine the direction a vessel, craft, or vehicle is pointed (heading), but rather the direction of the motion of the GPS antenna relative to the ground.

The latency of the instantaneous Doppler velocity is always 0.15 seconds. The latency represents an estimate of the delay caused by the tracking loops under acceleration of approximately 1 G. For most users, the latency can be assumed to be zero (instantaneous velocity).

Message ID: **100**
Log Type: **Synch**

Recommended Input:

       log psrvela ontime 1

ASCII Example:

```
#PSRVELA,COM1,0,52.5,FINESTEERING,1337,403362.000,00000000,658B,
1984;SOL_COMPUTED,PSRDIFF,0.250,9.000,0.0698,26.582692,0.0172,
0.0*A94E5D48
```

> (i) Consider the case where vehicles are leaving a control center. The control center's coordinates are known but the vehicles are on the move. Using the control center's position as a reference, the vehicles are able to report where they are with PSRPOS and their speed and direction with PSRVEL at any time.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PSRVEL header	Log header		H	0
2	sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	vel type	Velocity type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+4

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
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 in seconds	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)	Ulong	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.105 PSRXYZ
Pseudorange Cartesian position and velocity

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the receiver's pseudorange position and velocity in ECEF coordinates. The position and velocity status field's indicate whether or not the corresponding data is valid. See *Figure 11, The WGS84 ECEF Coordinate System* on *page 422* for a definition of the ECEF coordinates.

The velocity status indicates varying degrees of velocity quality. To ensure healthy velocity, the velocity sol-status must also be checked. If the sol-status is non-zero, the velocity is likely invalid. It should be noted that the receiver does not determine the direction a vessel, craft or vehicle is pointed (heading) but rather the direction of the motion of the GNSS antenna relative to the ground.

The latency of the instantaneous Doppler velocity is always 0.15 seconds. The latency represents an estimate of the delay caused by the tracking loops under acceleration of approximately 1 G. For most users, the latency can be assumed to be zero (instantaneous velocity).

Message ID: 243
Log Type: Synch

Recommended Input:

>   log psrxyza ontime 1

ASCII Example:

```
#PSRXYZA,COM1,0,58.5,FINESTEERING,1419,340038.000,00000040,4A28,
2724;SOL_COMPUTED,SINGLE,-1634530.7002,-3664617.2823,4942495.5175,
1.7971,2.3694,2.7582,SOL_COMPUTED,DOPPLER_VELOCITY,0.0028,0.0231,
-0.0120,0.2148,0.2832,0.3297,"",0.150,0.000,0.000,12,12,0,0,0,06,
0,33*4FDBCDB1
```

> (i) The instantaneous Doppler is the measured Doppler frequency which consists of the satellite's motion relative to the receiver (Satellite Doppler + User Doppler) and the clock (local oscillator) drift.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PSRXYZ header	Log header		H	0
2	P-sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X σ	Standard deviation of P-X (m)	Float	4	H+32
8	P- Y σ	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z σ	Standard deviation of P-Z (m)	Float	4	H+40
10	V-sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H+44
11	vel type	Velocity type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+48
12	V-X	Velocity vector along X-axis (m/s)	Double	8	H+52
13	V-Y	Velocity vector along Y-axis (m/s)	Double	8	H+60
14	V-Z	Velocity vector along Z-axis (m/s)	Double	8	H+68
15	V-X σ	Standard deviation of V-X (m/s)	Float	4	H+76
16	V-Y σ	Standard deviation of V-Y (m/s)	Float	4	H+80
17	V-Z σ	Standard deviation of V-Z (m/s)	Float	4	H+84
18	stn ID	Base station ID	Char[4]	4	H+88
19	V-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+92
20	diff_age	Differential age in seconds	Float	4	H+96
21	sol_age	Solution age in seconds	Float	4	H+100
22	#SVs	Number of satellites tracked	Uchar	1	H+104
23	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+105
24			Char	1	H+106
25	Reserved		Char	1	H+107
26			Char	1	H+108
27	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+109
28	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+110
29	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+111
30	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+112
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.106 QZSSALMANAC
Decoded QZSS Almanac parameters

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the decoded almanac parameters as received from the satellite with the parity information removed and appropriate scaling applied.

> (i) For more information about QZSS almanac data, refer to the Interface Specifications for QZSS at http://qz-vision.jaxa.jp/USE/is-qzss/index_e.html.

Message ID: 1346
Log Type: Asynch

Recommended Input:

> log qzssalmanaca onchanged

ASCII Example:
```
#QZSSALMANACA,COM1,0,89.5,SATTIME,1642,148584.000,00000008,67D2,
39655;
1,
193,1642,208896.0,7.587582E-02,-2.94869425E-09,-1.4441238E+00,
-1.5737385E+00,1.7932513E+00,0.00000000,0.00000000,7.29336435E-05,
4.2159360E+07,7.11809030E-01,7,7*FB648921
```

> (i) The OEM6 family of receivers automatically saves almanacs in their Non-Volatile Memory (NVM), therefore creating an almanac boot file is not necessary.

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	QZSSALMANAC Header	Log header		H	0
2	#messages	Number of satellite PRN almanac messages to follow. Set to zero until almanac data is available	Ulong	4	H
3	PRN	Satellite PRN number for current message (dimensionless)	Ulong	4	H+4
4	week	Almanac reference week	Ulong	4	H+8
5	seconds	Almanac reference time (s)	Double	8	H+12
6	ecc	Eccentricity (dimensionless) defined for a conic section where $e = 0$ is a circle, $e = 1$ is a parabola, $0 < e < 1$ is an ellipse $e > 1$ is a hyperbola	Double	8	H+20

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
7	$\dot{\omega}$	Rate of right ascension (radians/s)	Double	8	H+28
8	ω_0	Right, ascension (radians)	Double	8	H+36
9	ω	Argument of perigee (radians) measurement along the orbital path from the ascending node to the point where the SV is closest to the Earth, in the direction of the SV's motion	Double	8	H+44
10	M_0	Mean anomaly of reference time (radians)	Double	8	H+52
11	a_{f0}	Clock aging parameter (s)	Double	8	H+60
12	a_{f1}	Clock aging parameter (s/s)	Double	8	H+68
13	N	Corrected mean motion (radians/s)	Double	8	H+76
14	A	Semi-major axis (m)	Double	8	H+84
15	inclination angle	Angle of inclination	Double	8	H+92
16	health-prn	SV health from Page 25 of subframe 4 or 5 (6 bits)	Ulong	4	H+100
17	health-alm	SV health from almanac (8 bits)	Ulong	4	H+104
18	Next PRN offset = H+4+(#messages x 104)				
19	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#messages x 104)
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.107 QZSSEPHEMERIS
Decoded QZSS parameters

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains a single set of QZSS ephemeris parameters.

Message ID: 1336
Log Type: Asynch

Recommended Input:

> log qzssephemerisa onchanged

ASCII Example:

```
#QZSSEPHEMERISA,COM1,0,93.5,SATTIME,1642,153690.000,00000008,1E9D,
39655;193,153690.000000000,7,201,201,1642,1642,154800.000000000,
4.216030971806980E+07,2.115802417E-09,-2.152109479,0.075863329,
-1.573817810,-0.000007546,0.000009645,-177.375000000,
-219.875000000,-0.000000797,-0.000002151,0.711859299,
-2.978695503E-10,-1.443966112,-1.636139580E-09,
713,154800.000000000,-5.122274160E-09,-0.000000163,1.250555215E-12,
0.000000000,FALSE,0.000072933,4.000000000,0,0,0,0*FBB52C7F
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	QZSSEPHEMERIS header	Log header		H	0
2	PRN	Satellite PRN number	Ulong	4	H
3	tow	Time stamp of subframe 0 (s)	Double	8	H+4
4	health	Health status - a 6-bit health code as defined in QZSS Interface Specification	Ulong	4	H+12
5	IODE1	Issue of ephemeris data 1	Ulong	4	H+16
6	IODE2	Issue of ephemeris data 2	Ulong	4	H+20
7	week	GPS reference week number	Ulong	4	H+24
8	z week	Z count week number. This is the week number from subframe 1 of the ephemeris. The 'toe week' (field #7) is derived from this to account for rollover	Ulong	4	H+28
9	toe	Reference time for ephemeris (s)	Double	8	H+32
10	A	Semi-major axis (m)	Double	8	H+40
11	ΔN	Mean motion difference (radians/s)	Double	8	H+48
12	M_0	Mean anomaly of reference time (radius)	Double	8	H+56
13	ecc	Eccentricity (dimensionless) quantity defined for a conic section where e = 0 is a circle, e = 1 is a parabola, 0<e<1 is an ellipse e>1 is a hyperbola	Double	8	H+64

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
14	ω	Argument of perigee (radians) measurement along the orbital path from the ascending node to the point where the SV is closest to the Earth, in the direction of the SV's motion	Double	8	H+72
15	cuc	Argument of latitude (amplitude of cosine, radians)	Double	8	H+80
16	cus	Argument of latitude (amplitude of sine, radians)	Double	8	H+88
17	crc	Orbit radius (amplitude of cosine, metres)	Double	8	H+96
18	crs	Orbit radius (amplitude of sine, metres)	Double	8	H+104
19	cic	Inclination (amplitude of cosine, radians)	Double	8	H+112
20	cis	Inclination (amplitude of sine, radians)	Double	8	H+120
21	I_0	Inclination angle at reference time (radians)	Double	8	H+128
22	$\dot{I}$	Rate of inclination angle (radians/s)	Double	8	H+136
23	ω_0	Right ascension (radians)	Double	8	H+144
24	$\dot{\omega}$	Rate of right ascension (radians/s)	Double	8	H+152
25	iodc	Issue of data clock	Ulong	4	H+160
26	toc	SV clock correction term s()	Double	8	H+164
27	tgd	Estimated group delay difference (s)	Double	8	H+172
28	a_{fo}	Clock aging parameter (s)	Double	8	H+180
29	a_{f1}	Clock aging parameter (s/s)	Double	8	H+188
30	a_{f2}	Clock aging parameter (s/s/s)	Double	8	H+196
31	AS	Anti-spoofing on: 0= FALSE 1=TRUE	Enum	4	H+204
32	N	Corrected mean motion (radians/s)	Double	8	H+208
33	URA	User Range Accuracy variance, m2. The ICD specifies that the URA index transmitted in the ephemerides can be converted to a nominal standard deviation value using an algorithm listed there. We publish the square of the nominal value (variance)	Double	8	H+216
34	Fit Interval	Curve fit interval: 0 = Ephemeris data are effective for 2 hours 1 = Ephemeris data are effective for more than 2 hours	Uchar	1	H+224
35	Reserved		Uchar	1	H+225
36	Reserved		Uchar	1	H+226
37	Reserved		Uchar	1	H+227
38	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+228
39	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.108 QZSSIONUTC
QZSS ionospheric and time information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the Ionospheric Model parameters (ION) and the Universal Time Coordinated parameters (UTC) for QZSS.

Message ID: 1347
Log Type: Asynch

Recommended Input:

        log qzssionutca onchanged

ASCII Example:
```
#QZSSIONUTCA,COM1,0,94.0,FINESTEERING,1642,153300.565,00480008,
158B,39655;1.396983861923218E-08,-6.705522537231444E-8,0.0000000000
00000E+000,1.788139343261719E-07,8.396800000000000E+04,7.5366400000
00000E+05,-7.864320000000000E+05,-6.946816000000000E+06,1642,307200
,-5.5879354476928711E-09,5.329070518E-15,1768,4,15,15,0*0204EEC1
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	QZSSIONUTC Header	Log header		H	0
2	a0	Alpha parameter constant term	Double	8	H
3	a1	Alpha parameter 1st order term	Double	8	H+8
4	a2	Alpha parameter 2nd order term	Double	8	H+16
5	a3	Alpha parameter 3rd order term	Double	8	H+24
6	b0	Beta parameter constant term	Double	8	H+32
7	b1	Beta parameter 1st order term	Double	8	H+40
8	b2	Beta parameter 2nd order term	Double	8	H+48
9	b3	Beta parameter 3rd order term	Double	8	H+56
10	utc wn	UTC reference week number	Ulong	4	H+64
11	tot	Reference time of UTC parameters	Ulong	4	H+68
12	A0	UTC constant term of polynomial	Double	8	H+72
13	A1	UTC 1st order term of polynomial	Double	8	H+80
14	wn lsf	Future week number	Ulong	4	H+88
15	dn	Day number (the range is 1 to 7 where Sunday=1 and Saturday=7)	Ulong	4	H+92
16	deltat ls	Delta time due to leap seconds	Long	4	H+96
17	deltat lsf	Future delta time due to leap seconds	Long	4	H+100
18	Reserved			4	H+104
19	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+108
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.109 QZSSRAWALMANAC
Raw QZSS almanac data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the undecoded almanac subframes as received from the QZSS satellite.

Message ID: 1345
Log Type: Asynch

Recommended Input:

> log qzssrawalmanaca onchanged

ASCII Example:

```
#QZSSRAWALMANACA,COM1,0,93.5,SATTIME,1642,153300.000,00480008,64C4,
39655;1642,208896.000,7,
1,8B000031C390C1820E33D007FEFE07CAE831C5293EBFE15049104A000001,
51,8B000031C613F3336A1FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000,
49,8B000031CD90F14E6A7CF3CF1CF1CF3CF3C73CF1CF1CF3CF3CF3CF000002,
50,8B000031CE14F24E6A0CF3CF1DF1CFFFFFFFFFFFFFFFFFFFFFFFFF000002,
56,8B000031D511F80FF70003292EF496000006FFFFFFFA4B6A0FE8040F0002,
52,8B000031E692F4A00A0FFF83F060F20801800820820820820820002080381,
53,8B000031E717F5808208208208208208208208208208208208208208208080*CA4
596F9
```

> (i) The OEM6 family of receivers automatically saves almanacs in their Non-Volatile Memory (NVM), therefore creating an almanac boot file is not necessary.

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	QZSSRAWALMANAC header	Log header		H	0
2	ref week	Almanac reference week number	Ulong	4	H
3	ref secs	Almanac reference time, in milliseconds (binary data) or seconds (ASCII data)	GPSec	4	H+4
4	#subframes	Number of subframes to follow	Ulong	4	H+8
5	svid	SV ID (satellite vehicle ID)[a]	Hex	2	H+12
6	data	Subframe page data	Hex	30	H+14
7	Next subframe offset = H+12+(#subframe x32)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+12+ (#subframes x 32)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. A value between 1 and 32 for the SV ID indicates the PRN of the satellite. Any other values indicate the page ID. SV ID 1 to 5 corresponds to QZSS PRN 193 to 197. Refer to QZSS Interface Specification for more details.

3.2.110 QZSSRAWEPHEM
QZSS Raw ephemeris information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw binary information for subframes one, two and three from the satellite with the parity information removed. Each subframe is 240 bits long (10 words - 24 bits each) and the log contains a total 720 bits (90 bytes) of information (240 bits x 3 subframes). This information is preceded by the PRN number of the satellite from which it originated. This message is not generated unless all 10 words from all 3 frames have passed parity.

Message ID: 1331
Log Type: Asynch

Recommended Input:

> log qzssrawephema onnew

ASCII Example:

```
#QZSSRAWEPHEMA,COM1,0,84.5,SATTIME,1642,230580.000,00000008,2F9E,
39655;193,1642,234000,8B00004B0F879AA01C8000000000000000000000F6DF3
921FE0005FFFDBD,8B00004B1009DFD2BB1EC493A98277E8FD26D924D5062DCAE8F
5B739210E,8B00004B108FFE5BC52864AE00591D003B8B02B6BFE13F3AF-
FE2AFDFF1E7*D2BD151E
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	QZSSRAWEPHEMAME header	Log header		H	0
2	prn	Satellite PRN number	Ulong	4	H
3	ref week	Ephemeris reference week number	Ulong	4	H+4
4	ref secs	Ephemeris reference time (s)	Ulong	4	H+8
5	subframe1	Subframe 1 data	Hex	30	H+12
6	subframe2	Subframe 2 data	Hex	30	H+42
7	subframe3	Subframe 3 data	Hex	30	H+72
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+102
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.111 QZSSRAWSUBFRAME
Raw QZSS subframe data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw QZSS subframe data.

A raw QZSS subframe is 300 bits in total, 10 words of 30 bits each. This includes the parity 6 bits at the end of each word, for a total of 60 parity bits. Note that in Field #4, the 'data' field below, the 60 parity bits are stripped out and only the raw subframe data remains, for a total of 240 bits. There are two bytes added onto the end of this 30 byte packed binary array to pad out the entire data structure to 32 bytes in order to maintain 4 byte alignment.

Message ID: 1330
Log Type: Asynch

Recommended Input:

> log qzssrawsubframea onnew

ASCII Example:

```
#QZSSRAWSUBFRAMEA,COM1,0,85.5,SATTIME,1642,230604.000,00000008,E56B
,39655;193,5,8B00004B11970637984EFBF7FD4D0FA10-
CA49631ACE140740A08FE0DFD43,65*6A7B9123
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	QZSSRAWSUBFRAME header	Log header		H	0
2	PRN	Satellite PRN number	Ulong	4	H
3	subfr id	Subframe ID	Ulong	4	H+4
4	data	Raw subframe data	Hex [30]	32[a]	H+8
5	chan	Signal channel number that the frame was decoded on	Ulong	4	H+40
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
7	[CR][LF]	Sentence terminator	-	-	-

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

3.2.112 RAIMSTATUS
RAIM status

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log provides information on Receiver Autonomous Integrity Monitoring (RAIM) status (refer to the RAIMMODE command on *page 252*).

Message ID: 1286
Log Type: Synch

Recommended Input:

> log raimstatusa ontime 1

ASCII Example:

```
#RAIMSTATUSA,COM1,0,93.5,FINESTEERING,1595,387671.500,00000008,
BF2D,5968;DEFAULT,PASS,NOT_AVAILABLE,0.000,NOT_AVAILABLE,0.000,
0*96A129EE
#RAIMSTATUSA,COM1,0,95.5,FINESTEERING,1595,387672.000,00000008,
BF2D,5968;APPROACH,PASS,PASS,17.037,PASS,25.543,0*2A53F2B9
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	RAIMSTATUS Header	Log header	-	H	0
2	RAIM Mode	RAIM mode (refer to *Table 51, RAIM Mode Types* on *page 253*)	Enum	4	H
3	Integrity status	Integrity Status (see *Table 121, Integrity Status* on *page 599*)	Enum	4	H+4
4	HPL status	Horizontal protection level status (see *Table 122, Protection Level Status* on *page 599*)	Enum	4	H+8
5	HPL	Horizontal protection level (m)	Double	8	H+12
6	VPL status	Vertical protection level status (see *Table 121, Integrity Status* on *page 599*)	Enum	4	H+20
7	VPL	Vertical protection level (m)	Double	8	H+24
8	#SVs	Number of excluded satellites	Ulong	4	H+32
9	System	Satellite system (see *Table 110, Satellite System* on *page 508*)	Enum	4	H+36

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
10	Satellite ID	In binary logs, the satellite ID field is 4 bytes. The 2 lowest order bytes, interpreted as a USHORT, are the system identifier. For instance, the PRN for GPS or the slot for GLONASS. The 2 highest-order bytes are the frequency channel for GLONASS, interpreted as a SHORT and zero for all other systems. In ASCII and abbreviated ASCII logs, the satellite ID field is the system identifier. If the system is GLONASS and the frequency channel is not zero, then the signed channel is appended to the system identifier. For example, slot 13, frequency channel -2 is output as 13-2	Ulong	4	H+40
11	Next offset field = H+36+(#SVs * 8)				
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+36 + (#SVs * 8)
13	[CR][LF]	Sentence terminator (ASCII only)			

Table 121: Integrity Status

Binary	ASCII	Description
0	NOT_AVAILABLE	RAIM is unavailable because either there is no solution or because the solution is unique, that is, there is no redundancy
1	PASS	RAIM succeeded. Either there were no bad observations or the bad observations were successfully removed from the solution
2	FAIL	RAIM detected a failure and was unable to isolate the bad observations

Table 122: Protection Level Status

Binary	ASCII	Description
0	NOT_AVAILABLE	When RAIM is not available for example, after issuing a FRESET command or when there are not enough satellites tracked to produce the required redundant observations
1	PASS	Current protection levels are below alert limits, meaning positioning accuracy requirements are fulfilled HPL < HAL VPL < VAL
2	ALERT	Current protection levels are above alert limits, meaning required positioning accuracy cannot be guaranteed by RAIM algorithm HPL ≥ HAL VPL ≥ VAL

3.2.113 RANGE
Satellite range information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The RANGE log contains the channel measurements for the currently tracked satellites. When using this log, please keep in mind the constraints noted along with the description.

It is important to ensure that the receiver clock has been set. This can be monitored by the bits in the *Receiver Status* field of the log header. Large jumps in pseudorange as well as Accumulated Doppler Range (ADR) occur as the clock is being adjusted. If the ADR measurement is being used in precise phase processing, it is important not to use the ADR if the "parity known" flag, in the *ch-tr-status* field, is not set as there may exist a half (1/2) cycle ambiguity on the measurement. The tracking error estimate of the pseudorange and carrier phase (ADR) is the thermal noise of the receiver tracking loops only. It does not account for possible multipath errors or atmospheric delays.

If multiple signals are being tracked for a given PRN, an entry for each signal, with the same PRN, appears in the RANGE logs. As shown in *Table 126, Channel Tracking Status* on *page 604*, these entries can be differentiated by bits 21-25, which indicate the signal type of the observation.

Message ID: **43**
Log Type: **Synch**

Recommended Input:

              log rangea ontime 30

Abbreviated ASCII Example:
```
<RANGE COM1 0 82.0 FINESTEERING 1729 155076.000 00004000 5103 11465
46
31 0 24514687.250 0.064 -128825561.494675 0.010 3877.473 45.0 563.310
18109C04
31 0 24514688.765 0.096 -100383546.734328 0.010 3021.415 39.8 558.900
02309C0B
14 0 20345286.178 0.047 -106915249.491005 0.008 90.799 47.6 10283.130
08109C24
14 0 20345282.367 0.130 -83310588.842026 0.008 70.753 44.0 10276.900
01303C2B
22 0 20789170.556 0.038 -109247823.573628 0.007 -1421.169 49.4 15829.450
18109C44
22 0 20789164.279 0.138 -85128150.759123 0.007 -1107.404 43.6 15822.400
11303C4B
11 0 21977065.699 0.057 -115490261.964920 0.009 1235.428 46.0 5831.400
18109C64
11 0 21977062.220 0.201 -89992401.903056 0.011 962.671 40.3 5823.900
11303C6B
1 0 23109644.678 0.073 -121441999.794897 0.011 2971.250 43.8 3239.620
18109CA4
1 0 23109646.769 0.073 -94630142.467139 0.011 2315.261 42.1 3233.420
02309CAB
1 0 23109647.385 0.009 -90687226.778371 0.009 2218.538 48.9 3237.080 01D03-
CA4
32 0 23839782.353 0.133 -125278916.608912 0.022 3033.561 38.7 2193.280
18109CC4
32 0 23839781.295 0.363 -97619939.025504 0.026 2363.815 35.1 2184.900
11303CCB
```

```
18 0 22923322.792 0.062 -120462840.747702 0.009 -2710.945 45.3 20493.260
18109D04
18 0 22923320.071 0.350 -93867119.471860 0.012 -2112.426 35.5 20484.400
11303D0B
24 0 23708761.188 0.111 -124590391.778428 0.015 -2376.459 40.2 10643.820
08109D24
24 0 23708763.572 0.065 -97083440.180816 0.015 -1851.788 43.1 10639.420
02309D2B
24 0 23708765.724 0.009 -93038305.697497 0.008 -1774.807 49.1 10641.680
01D03D24
19 0 23739234.067 0.078 -124750470.392697 0.013 -2778.561 43.3 12263.180
08109D64
19 0 23739230.131 0.250 -97208136.646475 0.014 -2165.115 38.4 12255.400
01303D6B
61 9 22189063.544 0.155 -118654856.801346 0.011 -3985.235 43.3 13310.882
08119E04
61 9 22189063.246 0.055 -92287085.024614 0.011 -3099.631 37.6 13303.964
00B13E0B
47 0 21209673.567 0.147 -113059527.680842 0.011 -804.710 43.8 7342.680
08119E24
47 0 21209679.575 0.043 -87935228.320976 0.011 -625.886 39.7 7334.968
00B13E2B
46 5 24097664.754 0.213 -128680178.570435 0.014 -3740.543 40.6 10098.600
08119E44
46 5 24097669.137 0.048 -100084595.729257 0.015 -2909.311 38.8 10082.838
10B13E4B
39 3 21484445.079 0.161 -114645140.076744 0.012 2864.162 43.0 4463.150
18119E64
39 3 21484447.532 0.046 -89168467.325722 0.013 2227.683 39.1 4453.468
10B13E6B
38 8 19445896.471 0.101 -103949483.524466 0.008 -389.973 47.1 11640.260
18119E84
38 8 19445897.101 0.048 -80849619.556577 0.009 -303.312 38.8 11632.974
00B13E8B
48 7 21301665.694 0.166 -113829687.684616 0.011 3143.656 42.8 3778.910
08119EA4
48 7 21301667.294 0.054 -88534230.502244 0.012 2445.068 37.8 3770.968
10B13EAB
54 11 20899591.029 0.131 -111837944.708346 0.009 -401.734 44.8 7155.190
18119EC4
54 11 20899589.241 0.024 -86985062.942139 0.009 -312.461 44.8 7146.970
10B13ECB
55 4 23127316.661 0.318 -123455195.443877 0.020 3067.787 37.1 1588.420
18119EE4
55 4 23127321.850 0.032 -96020732.562183 0.021 2386.060 42.3 1580.442
00B13EEB
12 0 26239080.161 0.048 -137887256.553732 0.015 -2696.802 47.6 11527.710
48539C24
12 0 26239085.285 0.012 -102967750.707625 0.013 -2013.883 46.8 11523.770
41933C24
12 0 26239083.219 0.011 -105653860.401460 0.013 -2066.457 47.3 11523.712
42333C24
12 0 26239094.196 0.019 -104310841.607718 0.014 -2040.204 42.7 11522.970
42933C24
11 0 25589806.061 0.045 -134475330.397885 0.013 -729.686 48.0 4974.653
48539C64
11 0 25589809.285 0.010 -100419891.315177 0.012 -545.179 47.8 4969.770
41933C64
11 0 25589806.124 0.010 -103039536.069621 0.011 -559.405 48.0 4969.734
42333C64
11 0 25589818.004 0.017 -101729751.744395 0.013 -552.305 43.7 4967.060
42933C64
8 0 39844800.850 0.077 -207482308.002186 0.018 -507.335 37.4 12048.980
18149C84
```

```
8 0 39844800.076 0.043 -160438471.200694 0.013 -392.547 42.5 12038.660
00349C84
```

> **617D**
> **FPk6D**
> For the OEM617D and FlexPak6D receivers, a RANGE_1 log can be requested to get RANGE data from the second antenna. As described in *Table 3, Binary Message Header Structure* on *page 26*, the message type indicates the log is from the second antenna. To request an ASCII log enter RANGEA_1, and for a binary log enter RANGEB_1.

(i) Consider the case where you have a computer to record GNSS data at a fixed location, and another laptop in the field also recording GNSS data as you travel. Can you take the difference between the recorded location and the known location of the fixed point and use that as an error correction for the recorded data in the field?

The simple answer is yes. You can take the difference between recorded position and known location and apply this as a position correction to your field data. Then, what is the difference between pseudorange and position differencing?

The correct and more standard way of computing this correction is to compute the range error to each GNSS satellite being tracked at your fixed location and to apply these range corrections to the observations at your mobile station.

The position corrections method is seldom used in industry. The drawback of this method is that computed corrections vary depending on the location of the fixed station. The geometry is not accounted for between the fixed station and the tracked satellites. Also, position corrections at the fixed site are computed with a certain group of satellites while the field station is tracking a different group of satellites. In general, when the position correction method is used, the farther the fixed and field stations are apart, the less accurate the solution.

The range corrections method is more commonly used in the industry. The advantage of using this method is that it provides consistent range corrections and hence field positions, regardless of the location of your fixed station. You are only able to obtain a "good" differential position if both the fixed and field stations are tracking the same four satellites at a minimum.

DGPS refers to using 1 base receiver, at a known location, and 1 or more rover receivers at unknown locations. As the position of the base is accurately known, we can determine the error present in GNSS at any given instant by either of the two methods previously described. We counter the bias effects present in GNSS including: ionospheric, tropospheric, ephemeris, receiver and satellite clock errors. You could choose either method depending on your application and the accuracy required.

Table 123: Tracking State

State	Description	State	Description
0	Idle	6	Channel steering
1	Sky Search	7	Frequency lock loop

2	Wide frequency band pull-in	9	Channel alignment
3	Narrow frequency band pull-in	10	Code search
4	Phase lock loop	11	Aided phase lock loop

Table 124: Correlator Type

State	Description
0	N/A
1	Standard correlator: spacing = 1 chip
2	Narrow Correlator®: spacing < 1 chip
3	Reserved
4	Pulse Aperture Correlator (PAC)
5-6	Reserved

Table 125: Channel Tracking Example

	N7		N6		N5		N4		N3		N2		N1	N0
0x	0		8		1		0		9		C		0	4
Bit #	31 | 30 29 28	27	26 25 24	23 22 21	20	19 | 18 17 16	15 14 13	12	11	10	9 8	7 6 5	4	3 2	1 0
Binary[a]	0 | 0 0 0	1	0 0 0	0 0 0	1	0 | 0 0 0	1 0 0	1	1	1	0 0	0 0 0	0	0 1	0 0
Data	Chan. Assignment | Phase Measurement | Reserved (R) | Lock Flag	Primary L1	Reserved (R)	Signal Type	Grouping	Reserved (R)	Satellite System	Correlator Spacing	Code locked flag	Parity flag	Phase lock flag	Channel Number	Tracking State
Value	Automatic | Half Cycle | Reserved (R) | Lock Out	Primary	Reserved (R)	L1 C/A	Grouped	Reserved (R)	GPS	PAC	Locked	Known	Locked	Channel 0	L1 Phase Lock Loop

a. For a complete list of hexadecimal and binary equivalents please refer to the Unit Conversions page on our website at www.novatel.com through Support | Knowledge and Learning.

Table 126: Channel Tracking Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Tracking state	0-11, see *Table 123, Tracking State* on *page 602*
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	SV channel number	(n-1) (0 = first, n = last) n depends on the receiver
	5	0x00000020		
	6	0x00000040		
	7	0x00000080		
N2	8	0x00000100		
	9	0x00000200		
	10	0x00000400	Phase lock flag	0 = Not locked, 1 = Locked
	11	0x00000800	Parity known flag	0 = Not known, 1 = Known
N3	12	0x00001000	Code locked flag	0 = Not locked, 1 = Locked
	13	0x00002000	Correlator type	0-7, see *Table 124, Correlator Type* on *page 603*
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000	Satellite system	0 = GPS 1= GLONASS 2 = SBAS 3 = Galileo 4 = BeiDou 5 = QZSS 6 = Reserved 7 = Other
	17	0x00020000		
	18	0x00040000		
	19	0x00080000	Reserved	

Nibble #	Bit #	Mask	Description	Range Value
N5	20	0x00100000	Grouping	0 = Not grouped, 1 = Grouped
	21	0x00200000	Signal type	Dependent on satellite system above: GPS: GLONASS: 0 = L1 C/A 0 = L1 C/A 5 = L2 P 1 = L2 C/A 9 = L2 P codeless 5 = L2 P 14 = L5 Q 17 = L2 C Galileo: QZSS: 1 = E1B 0 = L1 C/A 2 = E1C 14 = L5Q 12 = E5a Q 17 = L2C 17 = E5b Q 20 = AltBOC Q SBAS: Other: 0 = L1 C/A 19 = L-Band 6 = L5I BeiDou 0 = B1 with D1 data 1 = B2 with D1 data 4 = B1 with D2 data 5 = B2 with D2 data
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000	Reserved	
	27	0x08000000	Primary L1 channel	0 = Not primary, 1 = Primary
N7	28	0x10000000	Carrier phase measurement [a]	0 = Half Cycle Not Added 1 = Half Cycle Added
	29	Reserved		
	30	0x40000000	PRN lock flag [b]	0 = PRN Not Locked Out 1 = PRN Locked Out
	31	0x80000000	Channel assignment	0 = Automatic, 1 = Forced

a. This bit is zero until the parity is known and the parity known flag (bit 11) is set to 1.

b. A PRN can be locked out using the LOCKOUT command.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RANGE header	Log header		H	0
2	# obs	Number of observations with information to follow [a]	Ulong	4	H
3	PRN/slot	Satellite PRN number of range measurement (GPS: 1 to 32, SBAS: 120 to 138 or 183-187, GLONASS: 38 to 61 (see GLONASS Slot and Frequency Numbers section of this manual) and Galileo: 1 to 36, QZSS: 193-197, BDS: 1-30	Ushort	2	H+4
4	glofreq	(GLONASS Frequency + 7) (see GLONASS Slot and Frequency Numbers section of this manual)	Ushort	2	H+6
5	psr	Pseudorange measurement (m)	Double	8	H+8
6	psr std	Pseudorange measurement standard deviation (m)	Float	4	H+16
7	adr	Carrier phase, in cycles (accumulated Doppler range)	Double	8	H+20
8	adr std	Estimated carrier phase standard deviation (cycles)	Float	4	H+28
9	dopp	Instantaneous carrier Doppler frequency (Hz)	Float	4	H+32
10	C/No	Carrier to noise density ratio $C/No = 10[\log_{10}(S/N_0)]$ (dB-Hz)	Float	4	H+36
11	locktime	# of seconds of continuous tracking (no cycle slipping)	Float	4	H+40
12	ch-tr-status	Tracking status (see *126, Channel Tracking Status* on *page 604* and the example in *Table 126*)	Ulong	4	H+44
13...	Next PRN offset = H + 4 + (#obs x 44)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#obs x 44)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. Satellite PRNs may have multiple lines of observations, one for the L1, L2 and L5 frequencies. For Galileo, there may be one for the E1, E5a, E5b and E5AltBoc frequencies.

3.2.114 RANGECMP
Compressed version of the RANGE log

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the RANGE data in a compressed format.

Message ID: **140**
Log Type: **Synch**

Recommended Input:

> log rangecmpa ontime 10

ASCII Example:
```
#RANGECMPA,COM1,0,63.5,FINESTEERING,
1429,226780.000,00000000,9691,2748;
26,
049C10081857F2DF1F4A130BA2888EB9600603A709030000,
0B9C3001225BF58F334A130BB1E2BED473062FA609020000,
449C1008340400E0AAA9A109A7535BAC2015CF71C6030000,
4B9C300145030010A6A9A10959C2F09120151F7166030000,
...
0B9D301113C8FFEFC284000C6EA051DBF3089DA1A0010000,
249D1018C6B7F67FA228820AF2E5E39830180AE1A8030000,
2B9D301165C4F8FFB228820A500A089F31185FE0A8020000,
449D1018BE18F41F2AACAD0A1A934EFC40074ECF88030000,
4B9D301182B9F69F38ACAD0A3E3AC28841079FCB88020000,
849D101817A1F95F16D7AF0A69FBE1FA401D3FD064030000,
8B9D30112909FB2F20D7AF0A9F24A687521DDECE64020000,
249E1118AF4E0470F66D4309A0A631CD642CF5B821320000,
2B9EB110A55903502F6E4309EE28D1AD032C7CB7E1320000,
849E1118B878F54F4ED2AA098C35558A532BDE1765220000,
8B9EB110ABCFF71F5ED2AA09CB6AD0F9032B9D16C5220000*0EEEAD18
```

617D

FPk6D

For the OEM617D and FlexPak6D receivers, a RANGECMP_1 log can be requested to get RANGECMP data from the second antenna. As described in *Table 3, Binary Message Header Structure* on *page 26*, the message type indicates the log is from the second antenna. To request an ASCII log enter RANGECMPA_1, and for a binary log enter RANGECMPB_1.

> (i) Consider the case where commercial vehicles are leaving a control center. The control center's coordinates are known but the vehicles are on the move. Using the control center's position as a reference, the vehicles are able to report where they are at any time. Post-processed information gives more accurate comparisons.
>
> Post-processing can provide post mission position and velocity using raw GNSS collected from the vehicles. The logs necessary for post-processing include:
>
> RANGECMPB ONTIME 1
>
> RAWEPHEMB ONCHANGED
>
> This is an example of data collection for post-processing. OEM6 based output is compatible with post-processing software from NovAtel's Waypoint Products.Refer to our website at www.novatel.com/support/ for more details.

Table 127: Range Record Format (RANGECMP only)

Data	Bit(s) first to last	Length (bits)	Scale Factor	Units
Channel Tracking Status	0-31	32	see *Table 126, Channel Tracking Status* on *page 604*	-
Doppler Frequency	32-59	28	1/256	Hz
Pseudorange (PSR)	60-95	36	1/128	m
ADR [a]	96-127	32	1/256	cycles
StdDev-PSR	128-131	4	See note[b]	m
StdDev-ADR	132-135	4	(n+1)/512	cycles
PRN/Slot [c]	136-143	8	1	-
Lock Time [d]	144-164	21	1/32	s
C/No [e]	165-169	5	(20 + n)	dB-Hz
Reserved	170-191	22		

a. ADR (Accumulated Doppler Range) is calculated as follows:

ADR_ROLLS = (RANGECMP_PSR / WAVELENGTH + RANGECMP_ADR) / MAX_VALUE

Round to the closest integer

IF (ADR_ROLLS $\leq$ 0)

      ADR_ROLLS = ADR_ROLLS - 0.5

ELSE

      ADR_ROLLS = ADR_ROLLS + 0.5

At this point integerise ADR_ROLLS

CORRECTED_ADR = RANGECMP_ADR - (MAX_VALUE*ADR_ROLLS)

where

ADR has units of cycles

WAVELENGTH = 0.1902936727984 for GPS L1

WAVELENGTH = 0.2442102134246 for GPS L2

MAX_VALUE = 8388608

Note: GLONASS satellites emit L1 and L2 carrier waves at a satellite-specific frequency, refer to the GLONASS section of our book <u>An Introduction to GNSS</u>, available on our website at <u>www.novatel.com/support/</u>.

b.

Code	StdDev-PSR (m)
0	0.050
1	0.075
2	0.113
3	0.169
4	0.253
5	0.380
6	0.570
7	0.854
8	1.281
9	2.375
10	4.750
11	9.500
12	19.000
13	38.000
14	76.000
15	152.000

c. GPS: 1 to 32, SBAS: 120 to 138 or 183-187, Galileo: 1 to 36, QZSS: 193-197, and GLONASS: 38 to 61, (see GLONASS Slot and Frequency Numbers section in this manual) and BDS: 1-30.

d. The *Lock Time* field of the RANGECMP log is constrained to a maximum value of 2,097,151 which represents a lock time of 65535.96875 s (2097151 ¸ 32).

e. C/No is constrained to a value between 20-51 dB-Hz. Thus, if it is reported that C/No = 20 dB-Hz, the actual value could be less. Likewise, if it is reported that C/No = 51, the true value could be greater.

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	RANGECMP header	Log header		H	0
2	#obs	Number of satellite observations with information to follow	Ulong	4	H
3	1st range record	Compressed range log in format of *Table 127* on *page 608*	Hex	24	H+4
4	Next rangecmp offset = H+4 (#obs x 24)				
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#obs x 24)
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.115 RANGECMP2
Compressed version of the RANGE log

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the RANGE data in a compressed format to handle more channels and different channel types than the RANGECMP log.

Message ID: 1273

Log Type: Synch

Recommended Input:

> log rangecmp2a ontime 10

Example:

```
#RANGECMP2A,COM1,0,84.5,FINESTEERING,1681,163457.000,00000020,1FE3,
10526;634,000D00F4FDDF05920620E1FFFF2979E806E81301C8FFE4FFFF03106B5
A50A902C8FF01100054F6BD05410720E1FFFF2996EA0E90FB01E2FFE4FFFF030E0D
65681603E3FF020400ACDCD605C40320E1FFFF697B080E9859801300E4FFFF4310C
94FB8C70114000317002C554685260520E1FFFF295F4412B0AD03C4FFE4FFFF03D5
A60D18C705C4FF0401008452B08583F92FE1FFFF2998AC65302C800000E4FFFF03F
32EDF784B0000000520000C8500056CFD2FE1FFFF295FA40DD04A822300E4FFFF03
B8242A58F8022300061F00C0081385EFFB2FE1FFFF295FC408A83884F8FFE4FFFF0
3B8861608C286F8FF081E008CB25105970520E1FFFF295C2604989483CEFFE4FFFF
03F2862F489006CFFF0914003027E204930020E1FFFF695E4407188602DDFFE4FFF
F43B8241480C903DDFF0A0E0050E3E305D3F92FE1FFFF2979C89C506D800700E4FF
FF030F4BDD603A8006000B1900D8F3CC8543FB2FE1FFFF297A280950F2002500E4F
FFF03F1286880E8022500140118341C0F0581F92FE1FFFF299D4404D02401F2FFE4
FFFF03920C2F900D82FAFF160D158CFA6B85400820E1FFFF69BAA600B83D02D9FFE
4FFFF03734A4380EA04CEFF170B178874EF0409FA2FE1FFFF299D6409D01904E6FF
E4FFFF0374EA31304D87DAFF180213C8039884FD0020E1FFFF697FE401007082D4F
FE4FFFF033B0616688084C4FF19131A5CDC9585F9FE2FE1FFFF69B8C80E08E58002
00E4FFFF0357C830A8D001EBFF1B0C16A45CA384C80220E1FFFF697F6401888A04E
FFFE4FFFF033A463D605E8802001C031C905434051D0720E1FFFF299CC60B18E881
F3FFE4FFFF0339462D38E182FBFF231050F05E6406B9FD1FE6FFFF6998080F10138
01300*61B80516
```

> **617D**
> **FPk6D**
> For the OEM617D and FlexPak6D receivers, a RANGECMP2_1 log can be requested to get RANGECMP2 data from the second antenna. As described in *Table 3, Binary Message Header Structure* on *page 26*, the message type indicates the log is from the second antenna. To request an ASCII log enter RANGECMP2A_1, and for a binary log enter RANGECMP2B_1.

Field #	Field Type	Description	Format	Binary Bytes	Binary Offset
1	RANGECMP2 header	Log Header		H	0

Field #	Field Type	Description	Format	Binary Bytes	Binary Offset
2	# bytes	Number of bytes in the compressed binary range data[a]	Uchar	4	H
3	RangeData	Compressed binary range data in the format of [b] *Table 128, Satellite Block of the Range Record Format (RANGECMP2 only)* on *page 612* and *Table 129, Signal Block of the Range Record Format (RANGECMP2 only)* on *page 613*	Uchar	#bytes	H+4
4	xxxx	32-bit CRC (ASCII and binary only)	Hex	4	H+4+ (# bytes)
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

　a.　Maximum is 2880 bytes for 120 channels; maximum 5760 for 240 channels.

　b.　The compressed binary range data is organized into satellite blocks, one for each satellite. Each satellite block is followed by a variable number of signal blocks corresponding to the same satellite. For example, a Satellite Block for GPS PRN 17 may be followed by two Signal Blocks for the L1 C/A and L2C signals.

Table 128: Satellite Block of the Range Record Format (RANGECMP2 only)

Data	Description	Bit(s) first to last	Length (bits)	Scale Factor	Units
SV Channel Number	Receiver SV channel number	0-7	8	-	-
Satellite Identifier	Satellite identifier specific to the satellite system: GPS: 1-32 SBAS: 120-138, 183-187 Galileo: 1-36 QZSS: 193-197 GLONASS: 1-24 BDS: 1-30	8-15	8	-	-
GLONASS Frequency Identifier	GLONASS frequency channel offset by +7	16-19	4	(7+Ch#)	-
Satellite System Identifier	Defined in *Table 110* on *page 508*	20-24	5	-	Enum
Reserved		25	1	-	-
Pseudorange Base	Pseudorange base value to be combined with PSR Diff and Phaserange Diff in each following Signal Block	26-54	29	1	m
Doppler Base	Doppler base value to be combined with the Scaled Doppler Diff value in each following Signal Block	55-75	21	1	Hz
Number of Signal Blocks	The number of Signal Blocks to follow this Satellite Block. See *Table 129* for Signal Block definition	76-79	4	-	-

Table 129: Signal Block of the Range Record Format (RANGECMP2 only)

Data	Description	Bit(s) first to last	Length (bits)	Scale Factor	Units
Signal Type	Defined in *Table 133, Signal Type (only in RANGECMP2) on page 615*	0-4	5	-	Enum
Phase Lock	Phase Lock: 0 = Not locked, 1 = Locked	5	1	-	Bool
Parity Known	Parity Known: 0 = Not known, 1 = Known	6	1	-	Bool
Code Lock	Code Lock: 0 = Not locked, 1 = Locked	7	1	-	Bool
Locktime	Time of continuous tracking with no cycle slips. The locktime value saturates at a maximum of 131071ms	8-24	17	1	ms
Correlator Type	Correlator type: (see Table 124, Page 603)	25-28	4	-	Enum
Primary Signal	Primary signal: 0 = Not primary, 1 = Primary	29	1	-	Bool
Carrier Phase Measurement	Carrier phase measurement: 0 = Half cycle not added, 1 = Half cycle added	30	1	-	Bool
Reserved		31	1	-	-
C/No	Carrier to Noise density ratio	32-36	5	(20 n)	dB-Hz
StdDev PSR	Pseudorange Standard Deviation (defined *Table 130, Std Dev PSR Scaling on page 614*)	37-40	4	Bit Field in *Table 130*	-
StdDev ADR	Carrier-Phase Standard Deviation (defined *Table 131, Std Dev ADR Scaling on page 614*)	41-44	4	Bit Field in *Table 131*	-
PSR Diff	Pseudorange Diff to be combined with Pseudorange base i.e., PSR = PSRBase + PSRDiff/128	45-58	14	1/128	m (unsigned)
Phaserange Diff	Phaserange Diff to be combined with Pseudorange Base i.e., ADR = PSRBase + PhaserangeDiff/2048	59-78	20	1/2048	m (unsigned)
Scaled Doppler Diff	Doppler Diff to be combined with Doppler Base. Note that all Doppler values are scaled to the L1/E1 equivalent value. (refer to *Table 132, L1/E1/B1 Scaling on page 614*) i.e., Doppler = (DopplerBase + ScaledDopplerDiff/256)/L1ScaleFactor	79-95	17	1/256	Hz (signed)

Table 130: Std Dev PSR Scaling

PSR Std Dev Bit Field Value	Represented Std Dev (m)	PSR Std Dev Bit Field Value	Represented Std Dev (m)
0	0.02	8	0.491
1	0.03	9	0.732
2	0.045	10	1.092
3	0.066	11	1.629
4	0.099	12	2.43
5	0.148	13	3.625
6	0.22	14	5.409
7	0.329	15	>5.409

Table 131: Std Dev ADR Scaling

ADR Std Dev Bit Field Value	Represented Std Dev (cycles)	ADR Std Dev Bit Field Value	Represented Std Dev (Cycles)
0	0.00391	8	0.03933
1	0.00521	9	0.05249
2	0.00696	10	0.07006
3	0.00929	11	0.09350
4	0.01239	12	0.12480
5	0.01654	13	0.16656
6	0.02208	14	0.22230
7	0.02947	15	>0.22230

Table 132: L1/E1/B1 Scaling

Satellite System	Signal Type	L1/E1/B1 Scale Factor
GPS	L1CA	1.0
	L2Y	154/120
	L2C	154/120
	L5Q	154/115
GLONASS	L1CA	1.0
	L2CA	9/7
	L2P	9/7
SBAS	L1CA	1.0
	L5I	154/115
Galileo	E1	1.0
	E5A	154/115
	E5B	154/118
	AltBOC	4

Satellite System	Signal Type	L1/E1/B1 Scale Factor
QZSS	L1CA	1.0
	L2C	154/120
	L5Q	154/115
LBAND	LBAND	1.0
BDS	B1	1.0
	B2	1526/1180

Table 133: Signal Type (only in RANGECMP2)

Satellite System	Signal Type	Value
GPS	L1CA	1
	L2Y	4
	L2CM	5
	L5Q	7
GLONASS	L1CA	1
	L2CA	3
	L2P	4
SBAS	L1CA	1
	L5I	2
Galileo	E1C	1
	E5AQ	2
	E5BQ	3
	AltBOCQ	4
QZSS	L1CA	1
	L2CM	3
	L5Q	4
LBAND	LBAND	1
BDS	B1D1I	1
	B1D2I	2
	B2D1I	3
	B2D2I	4

3.2.116 RANGEGPSL1
L1 version of the RANGE log

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log is identical to the RANGE log, see *page 600,* except that it only includes L1 GPS observations.

Message ID:　　　　**631**　　　　　　　　　**Log Type: Synch**

Recommended Input:

　　　　　log rangegpsl1a ontime 30

ASCII Example:

```
#RANGEGPSL1A,COM1,0,57.0,FINESTEERING,1337,404766.000,00000000,
5862,1984;
10,
14,0,21773427.400,0.037,-114420590.433332,0.006,-2408.171,49.9,
14963.280,18109C04,
22,0,24822942.668,0.045,-130445851.055756,0.009,-3440.031,48.0,
22312.971,08109C24,
25,0,20831000.299,0.033,-109468139.214586,0.006,1096.876,50.7,
7887.840,08109C44,
1,0,20401022.863,0.032,-107208568.887106,0.006,-429.690,51.1,
10791.500,18109C64,
24,0,23988223.932,0.074,-126058964.619453,0.013,2519.418,43.8,
493.550,18109C84,
11,0,22154466.593,0.043,-116423014.826717,0.007,-1661.273,48.4,
11020.952,08109CA4,
5,0,24322401.516,0.067,-127815012.260616,0.012,-1363.596,44.6,
6360.282,18109CC4,
20,0,22294469.347,0.043,-117158267.467388,0.008,2896.813,48.5,
4635.968,08109CE4,
30,0,23267589.649,0.051,-122271969.418761,0.009,822.194,47.0,
4542.270,08109D04,
23,0,24975654.673,0.058,-131247903.805678,0.009,3395.097,45.9,
406.762,18109D24*BE4B7D70
```

> Since the RANGEGPSL1 log includes only L1 GPS observations, it is smaller in size than the RANGE log which contains entries for multiple systems and signals. Use the RANGEGPSL1 log when data throughput is limited and you are only interested in GPS L1 range data. For GPS L1 only models, RANGE and RANGEGPSL1 logs are identical.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RANGEGPSL1 header	Log header		H	0
2	# obs	Number of L1 observations with information to follow	Long	4	H
3	PRN	Satellite PRN number of range measurement (GPS: 1 to 32)	Ushort	2	H+4
4	Reserved		Ushort	2	H+6
5	psr	Pseudorange measurement (m)	Double	8	H+8
6	psr std	Pseudorange measurement standard deviation (m)	Float	4	H+16
7	adr	Carrier phase, in cycles (accumulated Doppler range)	Double	8	H+20
8	adr std	Estimated carrier phase standard deviation (cycles)	Float	4	H+28
9	dopp	Instantaneous carrier Doppler frequency (Hz)	Float	4	H+32
10	C/No	Carrier to noise density ratio $C/No = 10[\log_{10}(S/N_0)]$ (dB-Hz)	Float	4	H+36
11	locktime	Number of seconds of continuous tracking (no cycle slipping)	Float	4	H+40
12	ch-tr-status	Tracking status (see *Table 126, Page 604*)	Ulong	4	H+44
13...	Next PRN offset = H + 4 + (#obs x 44)				
14	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#obs x 44)
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.117 RAWALM
Raw Almanac data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the undecoded GPS almanac subframes as received from the satellite. For more information about Almanac data, refer to our book [An Introduction to GNSS](#), available on our website at [www.novatel.com/support/](#).

Message ID: **74**
Log Type: **Asynch**

Recommended Input:

        log rawalma onchanged

ASCII Example:
```
#RAWALMA,COM1,0,56.0,SATTIME,1337,405078.000,00000000,CC1B,1984;
1337,589824.000,43,
3,8B04E4839F35433A5590F5AEFD3900A10C9AAA6F40187925E50B9F03003F,
27,8B04E483A1325B9CDE9007F2FD5300A10DA5562DA3ADC0966488DD01001A,
4,8B04E483A1B44439979006E2FD4F00A10D15D96B3B021E6C6C5F23FEFF3C,
28,8B04E483A3B05C5509900B7CFD5800A10CC483E2BFA1D2613003BD050017,
5,8B04E483A43745351C90FCB0FD4500A10D8A800F0328067E5DF8B6100031,
57,8B04E483A6337964E036D74017509F38E13112DF8DD92D040605EEAAAAAA,
6,8B04E483A6B54633E390FA8BFD3F00A10D4FACBC80B322528F62146800BA,
29,8B04E483A8B05D47F7901B20FD5700A10CE02D570ED40A0A2216412400CB,
7,8B04E483A935476DEE90FB94FD4300A10D93ABA327B7794AE853C02700BA,
.
.
.
1,8B04E483D8B641305A901B9DFD5A00A10CE92F48F1BA0A5DCCCB7500003B,
25,8B04E483DAB25962259004FCFD4C00A10DC154EEE5C555D7A2A5010D000D,
2,8B04E483DB37424AA6900720FD4F00A10C5AD89BAA4DC1460790B6FC000F,
26,8B04E483DD305A878C901D32FD5B00A10C902EB7F51DB6B6CE95C701FFF4*83C
AE97A
```

> (i) The OEM6 family of receivers automatically saves almanacs in their Non-Volatile Memory (NVM), therefore creating an almanac boot file is not necessary.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RAWALM header	Log header		H	0
2	ref week	Almanac reference week number	Ulong	4	H
3	ref secs	Almanac reference time (ms)	GPSec	4	H+4
4	#subframes	Number of subframes to follow	Ulong	4	H+8
5	svid	SV ID (satellite vehicle ID) [a]	Ushort	2	H+12
6	data	Subframe page data	Hex	30	H+14
7...	Next subframe offset = H+12+(#subframe x 32)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+12+ (#subframes x 32)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. A value between 1 and 32 for the SV ID indicates the PRN of the satellite. Any other values indicate the page ID. See section 20.3.3.5.1.1, *Data ID and SV ID*, of ICD-GPS-200C for more details. To obtain copies of ICD-GPS-200, refer to the Standards and References section of our website www.novatel.com/support/.

3.2.118 RAWCNAVFRAME
Raw CNAV frame data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log provides raw frame data from signals which contain the CNAV message (L2C, L5).

> ⓘ   The RAWCNAVFRAME log is not output by default. To receive this log, data decoding for L2C or L5 must be enabled using the `DATADECODESIGNAL` command (see *page 117*) for the specific signal.

Message ID: **1066**
Log Type: **Asynch**

Recommended Input:

       log rawcnavframea onnew

ASCII Example:

```
#RAWCNAVFRAMEA,COM1,88,84.0,SATTIME,1693,242208.000,00000020,EE56,1
0585;50,193,11,8B04B4ED919863A6671F473A31412695EFF3C026C0209FF07D60
1F775FEFE1FF987800000000*CC38FB17
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RAWCNAVFRAME header	Log header		H	0
2	sigchannum	Signal channel providing the bits	Ulong	4	H
3	PRN	Satellite PRN number	Ulong	4	H+4
4	frameId	frame ID	Ulong	4	H+8
5	data	Raw frame data	Hex[38]	38	H+12
6	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+50
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.119 *RAWEPHEM*
Raw ephemeris

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw binary information for subframes one, two and three from the GPS satellite L1 C/A signal with the parity information removed. Each subframe is 240 bits long (10 words - 24 bits each) and the log contains a total 720 bits (90 bytes) of information (240 bits x 3 subframes). This information is preceded by the PRN number of the satellite from which it originated. This message is not generated unless all 10 words from all 3 frames have passed parity.

Ephemeris data whose Time of Ephemeris (TOE) is older than six hours is not shown. Multiple logs are output, one for each GPS satellite with collected ephemeris information.

Message ID: 41
Log Type: Asynch

Recommended Input:

log rawephema onnew

ASCII Example:

```
#RAWEPHEMA,COM1,15,60.5,FINESTEERING,1337,405297.175,00000000,97B7,
1984;
3,1337,403184,8B04E4818DA44E50007B0D9C05EE664FFBFE695DF763626F00001
B03C6B3,8B04E4818E2B63060536608FD8CDAA051803A41261157EA10D2610626F3
D,8B04E4818EAD0006AA7F7EF8FFDA25C1A69A14881879B9C6F-
FA79863F9F2*0BB16AC3
  .
  .
  .
#RAWEPHEMA,COM1,0,60.5,SATTIME,1337,405390.000,00000000,97B7,1984;
1,1337,410400,8B04E483F7244E50011D7A6105EE664FFBFE695DF9E1643200001
200AA92,8B04E483F7A9E1FAAB2B16A27C7D41FB5C0304794811F7A10D40B564327
E,8B04E483F82C00252F57A782001B282027A31C0FBA0FC525F-
FAC84E10A06*C5834A5B
```

(i) A way to use only one receiver and achieve better than 1 metre accuracy is to use precise orbit and clock files. Three types of GPS ephemeris, clock and earth orientation solutions are compiled by an elaborate network of GNSS receivers around the world all monitoring the satellite characteristics. IGS rapid orbit data is processed to produce files that correct the satellite clock and orbit parameters. Since there is extensive processing involved, these files are available on a delayed schedule from the US National Geodetic Survey at: www.ngs.noaa.gov/orbits

Precise ephemeris files are available today to correct GPS data which was collected a few days ago. All you need is one GNSS receiver and a computer to process on. Replace the ephemeris data with the precise ephemeris data and post-process to correct range values.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RAWEPHEM header	Log header		H	0
2	prn	Satellite PRN number	Ulong	4	H
3	ref week	Ephemeris reference week number	Ulong	4	H+4
4	ref secs	Ephemeris reference time (s)	Ulong	4	H+8
5	subframe1	Subframe 1 data	Hex[30]	30	H+12
6	subframe2	Subframe 2 data	Hex[30]	30	H+42
7	subframe3	Subframe 3 data	Hex[30]	30	H+72
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+102
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.120 RAWGPSSUBFRAME
Raw subframe data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw GPS subframe data.

A raw GPS subframe is 300 bits in total. This includes the parity bits which are interspersed with the raw data ten times in six bit chunks, for a total of 60 parity bits. Note Field #5, below, has these 60 parity bits stripped out and only the raw subframe data remains, for a total of 240 bits.

Message ID:	**25**
Log Type:	**Asynch**

Recommended Input:

        log rawgpssubframea onnew

ASCII Example:
```
#RAWGPSSUBFRAMEA,COM1,59,62.5,SATTIME,1337,405348.000,00000000,F690
,1984;2,22,4,8B04E483F3B17EE037A3732FE0FC8CCF074303EBDF2F6505F5AAAA
AAAAA9,2*41E768E4
...
#RAWGPSSUBFRAMEA,COM1,35,62.5,SATTIME,1337,405576.000,00000000,F690
,1984;4,25,2,8B04E48406A8B9FE8B364D786EE827FF2F062258840EA4A10E20B9
64327E,4*52D460A7
...
#RAWGPSSUBFRAMEA,COM1,0,62.5,SATTIME,1337,400632.000,00000000,F690,
1984;20,9,3,8B04E4826AADFF3557257871000A26FC34A31D7A300BEDE5F-
FA3DE7E06AF,20*55D16A4A
```

> (i) The RAWGPSSUBFRAME log can be used to receive the data bits with the parity bits stripped out. Alternately, you can use the RAWGPSWORD log to receive the parity bits in addition to the data bits.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RAWGPSSUBFRAME header	Log header		H	0
2	decode #	Frame decoder number	Long	4	H
3	PRN	Satellite PRN number	Ulong	4	H+4
4	subfr id	Subframe ID	Ulong	4	H+8
5	data	Raw subframe data	Hex[30]	32[a]	H+12
6	chan	Signal channel number that the frame was decoded on	Ulong	4	H+44
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+48
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

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

3.2.121 RAWGPSWORD
Raw navigation word

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This message contains the framed GPS raw navigation words. Each log contains a new 30 bit navigation word (in the least significant 30 bits), plus the last 2 bits of the previous word (in the most significant 2 bits). The 30 bit navigation word contains 24 bits of data plus 6 bits of parity. The GPS reference time stamp in the log header is the time the first bit of the 30 bit navigation word was received. Only navigation data that has passed parity checking appears in this log. One log appears for each PRN being tracked every 0.6 seconds if logged ONNEW or ONCHANGED.

Message ID: **407**
Log Type: **Asynch**

Recommended Input:

> log rawgpsworda onnew

ASCII Example:
```
#RAWGPSWORDA,COM1,0,58.5,FINESTEERING,1337,405704.473,00000000,
9B16,1984;14,7FF9F5DC*8E7B8721

...
#RAWGPSWORDA,COM1,0,57.0,FINESTEERING,1337,405783.068,00000000,
9B16,1984;1,93FEFF8A*6DD62C81

...
#RAWGPSWORDA,COM1,0,55.5,FINESTEERING,1337,405784.882,00000000,
9B16,1984;5,FFFFF8CE*A948B4DE
```

> (i) The RAWGPSWORD log can be used to receive the parity bits in addition to the data bits. Alternately, you can use the RAWGPSSUBFRAME log which already has the parity bits stripped out

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RAWGPSWORD header	Log header		H	0
2	PRN	Satellite PRN number	Ulong	4	H
3	nav word	Raw navigation word	Hex[4]	4	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+8
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.122 RAWLBANDFRAME
Raw L-Band frame data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw L-Band frame data. The RAWLBANDPACKET (*page 627*) is output for OmniSTAR and TerraStar tracking.

> (i) In addition to a NovAtel receiver with L-Band capability, use of a DGPS service is required. Refer to the OEM6 Family Installation and Operation User Manual (OM-20000128) available at www.novatel.com/support/.

Message ID:	**732**
Log Type:	**Asynch**

Recommended Input:

> log rawlbandframea onnew

ASCII Example:

```
#RAWLBANDFRAMEA,COM2,0,73.5,FINESTEERING,1295,152802.068,00000040,4
F80,34461;9,1A1E,600,F6,00,62,35,C8,CD,34,E7,6A,A1,37,44,8F,A8,24,7
1,90,D0,5F,94,2D,94,3C,74,9C,F0,12,A3,4C,A7,30,AA,B6,2E,27,DD,DC,24
,BA,D3,76,8D,76,D9,E7,83,1A,C8,81,B0,62,1C,69,88,23,70,2A,06,C0,FC,
F8,80,2C,72,F1,2E,6B,C2,5B,EC,03,70,D3,F3,FE,EF,37,3D,17,37,1B,CF,B
E,AF,D1,02,15,96,D1,F6,58,56,AC,BD,A3,11,12,D0,3D,11,27,8A,87,28,0C
,0F,52,70,B3,2F,0C,0C,62,2D,B8,69,6C,52,10,DF,7D,BB,08,D6,CA,A9,5E,
77,66,96,C2,A0,63,3B,98,34,BC,D5,47,64,E0,00,37,10,4A,F7,C1,B6,83,8
F,06,94,21,FF,B4,27,15,B0,60,40,02,B4,AF,9C,9D,C2,D4,EA,95,68,86,0F
,0A,9D,2D,36,52,68,65,B8,A2,0B,00,21,80,64,8A,72,FF,59,B7,79,B9,49,
FD,F5,3C,48,1C,2F,77,F1,B2,9E,58,0A,81,05,1F,00,7B,00,1E,68,C9,A3,1
2,56,B8,2A,32,DF,D9,EA,03,9B,16,C6,17,2F,33,B3,5F,C4,F9,D2,97,75,64
,06,52,A1,B2,3A,4B,69,E7,EB,0F,97,D3,E6,BF,DE,AF,37,C6,10,13,9B,DC,
C9,E3,22,80,78,3F,78,90,D5,9F,D3,5F,AF,1F,7A,75,EF,77,8E,DE,AC,00,3
2,2E,79,FB,3F,65,F3,4F,28,77,B4,6D,F2,6F,31,24,B2,40,76,37,27,BC,95
,33,15,01,76,D5,F1,C4,75,16,E6,C6,AB,F2,FE,34,D9,C3,55,85,61,49,E6,
A4,4E,8B,2A,60,57,8A,E5,77,02,FC,9C,7D,D4,40,4C,1D,11,3C,9B,8E,C3,7
3,D3,3C,0D,FF,18.
.
.
.
,7A,21,05,CB,12,F6,DD,C3,DF,69,62,F5,70*3791693B
```

> (i) The data signal is structured to perform well in difficult or foliated conditions, so the service is available consistently more and has a higher degree of service reliability.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RAWLBANDFRAME header	Log header		H	0
2	Service ID	L-Band Beam Service ID	Ushort	2	H
3	Reserved	Reserved bits	Ushort	2	H+2
4	# of Records	# of Records to Follow	Ulong	4	
5	data	Raw L-Band frame data	Hex[512]	1	H+4
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+516
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.123 RAWLBANDPACKET
Raw L-Band data packet

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw L-Band packet data. The RAWLBANDPACKET log is only output for OmniSTAR or TerraStar tracking.

Message ID: **733**
Log Type: **Asynch**

Recommended Input:

      log rawlbandpacketa onnew

ASCII Example:

#RAWLBANDPACKETA,COM2,0,77.0,FINESTEERING,1295,238642.610,01000040,
C5B1,34461;9,07,DE,3A,F9,DF,30,7B,0D,CB*7E5205A8

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RAWLBANDPACKET header	Log header		H	0
2	#recs	Number of records to follow	Ulong	4	H
3	data	Raw L-Band data packet	Hex[512]	1	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+#recs
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.124 RAWSBASFRAME
Raw SBAS frame data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the raw SBAS frame data of 226 bits (8-bit preamble, 6-bit message type and 212 bits of data but without a 24-bit CRC). Only frame data with a valid preamble and CRC are reported.

Message ID: 973
Log Type: Asynch

Recommended Input:

> log rawsbasframea onnew

ASCII Example:

```
#RAWSBASFRAMEA,COM1,0,91.0,SATTIME,1610,341534.000,00000000,58E4,
38637;32,133,4,C6115FFC00000C009FFC07004C089FFDFFDFFDFFDFFF957BBB6B
FFFFC0,32*5AFC5F95

#RAWSBASFRAMEA,COM1,0,91.0,SATTIME,1610,341535.000,00000000,58E4,
38637;32,133,2,53084007FF9FFFFFC03002C0000F0009FFC004005FFD6B961E39
B9FB80,32*DB5DFA62

#RAWSBASFRAMEA,COM1,0,91.0,SATTIME,1610,341535.000,00000000,58E4,
38637;35,135,2,53084007FF9FFFFFC03002C0000F0009FFC004005FFD6B961E39
B9FB80,35*B72FF2A0

. . .

#RAWSBASFRAMEA,COM1,0,90.0,SATTIME,1610,341539.000,00000000,58E4,
38637;34,138,3,9A0C4000009FFC009FFDFFC007FB9FFDFFC0000040315B9BB96F
B95680,34*CB050361
```

> (i) The RAWSBASFRAME log output contains all the raw data required for an application to compute its own SBAS correction parameters.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RAWSBASFRAME header	Log header		H	0
2	decode #	Frame decoder number	Ulong	4	H
3	PRN	SBAS satellite PRN number	Ulong	4	H+4
4	WAASmsg id	SBAS frame ID	Ulong	4	H+8
5	data	Raw SBAS frame data. There are 226 bits of data and 6 bits of padding	Hex[29]	32[a]	H+12
6	chan	Signal channel number that the frame was decoded on	Ulong	4	H+44
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+48
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.

3.2.125 REFSTATION
Base station position and health

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the ECEF Cartesian position of the base station as received through the RTCM, RTCMV3, RTCA or CMR message. It also features a time tag, the health status of the base station and the station ID. This information is set at the base station using the FIXPOSITION command (*page 154*) and the DGPSTXID (*page 125*) command. See *Figure 11, The WGS84 ECEF Coordinate System* on *page 422* for a definition of the ECEF coordinates.

The base station health, Field #6, may be one of 8 values (0 to 7). Values 0 through 5 indicate the scale factor that is multiplied with the satellite UDRE one-sigma differential error values. Below are values 0 to 5 and their corresponding UDRE scale factors:

0: 1 (Health OK)   1: 0.75   2: 0.5   3: 0.3   4: 0.2   5: 0.1

The base station health field only applies to RTCM base stations. A value of 6 means the base station transmission is not monitored and a value of 7 means that the base station is not working.

Message ID: **175**
Log Type: **Asynch**

Recommended Input:

      log refstationa onchanged

ASCII Example:
```
#REFSTATIONA,COM1,0,66.5,FINESTEERING,1364,490401.124,80000000,
4E46,2310;00000000,-1634532.443,-3664608.907,4942482.713,0,
RTCA,"AAAA"*1E2A0508
```

Table 134: Base Station Status

Bit #	Mask	Description	Bit = 0	Bit = 1
0	0x00000001	Validity of the base station	Valid	Invalid

Table 135: Station Type

Base Station Type (Binary) (ASCII)		Description
0	NONE	Base station is not used
1	RTCM	Base station is RTCM
2	RTCA	Base station is RTCA
3	CMR	Base station is CMR
4	RTCMV3	Base station is RTCMV3

> (i) The REFSTATION log can be used for checking the operational status of a remotely located base station. You can verify that the base station is operating properly without traveling to it. This is especially useful for RTK work on long baselines.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	REFSTATION header	Log header		H	0
2	status	Status of the base station information (see *Table 134*)	Ulong	4	H
3	x	ECEF X value (m)	Double	8	H+4
4	y	ECEF Y value (m)	Double	8	H+12
5	z	ECEF Z value (m)	Double	8	H+20
6	health	Base station health, see the 2nd paragraph on the previous page	Ulong	4	H+28
7	stn type	Station type (see *Table 135, Station Type* on *page 629*)	Enum	4	H+32
8	stn ID	Base station ID	Char[5]	8[a]	H+36
9	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
10	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.

3.2.126 *REFSTATIONINFO*
Base Station position information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This is an extended version of the REFSTATION log with latitude, longitude and ellipsoidal height of the base station in WGS84. In addition to the base station position, ARP height, antenna model name and antenna serial number are available if provided by the base station only through RTCMV3.

Message ID: 1325
Log Type: Asynch
Recommended Input:

> log refstationinfoa onchanged

> (i) ARP, Antenna model and Antenna serial number are set at the base station using BASEANTENNAMODEL command (*page 84*).

For example: BASEANTENNAMODEL 702GG NVH05410007 1 USER 0 0 1234

ASCII Example:

```
#REFSTATIONINFOA,USB1,0,89.5,EXACT,0,0.000,00000040,D38F,6782;
51.116375174,-114.038254922,1048.502830628,WGS84,1.234,0,RTCMV3,
"0","702GG","NVH05410007"*BEDF8ECE
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	REFSTATIONINFO header	Log header	H	0	
2	latitude	Latitude (degrees)	Double	8	H
3	longitude	Longitude (degrees)	Double	8	H+8
4	height	Ellipsoidal Height (m)	Double	8	H+16
5	datum	Datum ID number (WGS84) (refer to *Table 24, Datum Transformation Parameters* on *page 120*)	Enum	4	H+24
6	ARP height	Base Antenna ARP (m)	Float	4	H+28
7	health	Base Station Health, see *Table 134, Base Station Status* on *page 629*	Ulong	4	H+32
8	Ref Stn Type	Base Station Type, see (*Table 135, Station Type* on *page 629*)	Enum	4	H+36
9	stn ID	Base Station ID	Char[5]	8[a]	H+40
10	Ant Model	Base Antenna Model Name	Char[32]	32	H+48
11	Ant Serial	Base Antenna Serial Number	Char[32]	32	H+80
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+112
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.

3.2.127 ROVERPOS
Position using ALIGN

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

ALIGN generates distance and bearing information between a master and rover receiver. This log outputs the position information of the rover when using the ALIGN feature. This log can only be output from a Y ALIGN model and can be output at both Master and Rover ends.

▼ You must have an ALIGN capable receiver to use this log.

ⓘ
1. ALIGN is useful for obtaining the relative directional heading of a vessel/body, separation heading between two vessels/bodies, or heading information with moving base and pointing applications.
1. The log can be output at the Y model Rover only if it is receiving the RTCAREFEXT message from the Master. The log can be output at any Master if the Master is receiving HEADINGEXTB from the Rover. Refer to the NovAtel application note APN-048 for details on HEADINGEXT (available at www.novatel.com/support/).
2. ROVERPOS is dependent on the output frequency of the RTCAREFEXT message from the master to the rover.

Message ID: **1052**
Log Type: **Asynch**

Recommended Input:

> log roverposa onchanged

Example 1:

```
#ROVERPOSA,COM1,0,21.5,FINESTEERING,1544,340322.000,00000008,7453,
4655;SOL_COMPUTED,NARROW_INT,51.11605565964,-114.03854655975,
1055.8559,-16.9000,WGS84,0.0130,0.0122,0.0206,"RRRR",0.0,0.0,13,
12,12,11,0,0,0,0*635B3A1C
```

ⓘ Asynchronous logs, such as ROVERPOS, should only be logged ONCHANGED or ONNEW otherwise the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

Field #	Field Type	Field Description	Format	Binary Bytes	Binary Offset
1	ROVERPOS header	Log Header		H	0
2	sol stat	Solution Status, see *Table 85* on *page 408*	Enum	4	H
3	pos type	Position Type see *Table 84* on *page 407*	Enum	4	H+4
4	lat	Rover WGS84 Latitude in degrees	Double	8	H+8
5	long	Rover WGS84 Longitude in degrees	Double	8	H+16
6	hgt	Rover MSL Height in metres	Double	8	H+24
7	undulation	Undulation in metres	Float	4	H+32
8	datum id#	WGS84 (default) (refer to *Table 24, Datum Transformation Parameters* on *page 120*)	Enum	4	H+36
9	lat σ	Latitude Std in metres	Float	4	H+40
10	long σ	Longitude Std in metres	Float	4	H+44
11	hgt σ	Height Std in metres	Float	4	H+48
12	stn id	Rover ID (default = "RRRR")	Char[4]	4	H+52
13	Reserved		Float	4	H+56
14	Reserved		Float	4	H+60
15	#SVs	Number of satellite tracked	Uchar	1	H+64
16	#solnSVs	Number of satellite in solution	Uchar	1	H+65
17	#obs	Number of satellites above elevation mask angle	Uchar	1	H+66
18	#multi	Number of satellites above the mask angle with L2, B2	Uchar	1	H+67
19			Hex	1	H+68
20	Reserved		Uchar	1	H+69
21			Uchar	1	H+70
22			Uchar	1	H+71
23	xxxx		Hex	1	H+72
24	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

3.2.128 RTCA Standard Logs

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

RTCA1	**Differential GPS Corrections**	
Message ID:	**10**	

RTCAEPHEM   **Ephemeris and Time Information**
Message ID: **347**

RTCAOBS   **Base Station Observations**
Message ID: **6**

RTCAOBS2   **Base Station Observations**
Message ID: **805**

RTCAOBS3   **Base Station Observation for ALIGN**
MESSAGE ID: **1340**

RTCAREF   **Base Station Parameters**
Message ID: **11**

RTCAREFEXT   **Extended Base Station Parameters for ALIGN**
Message ID: **1049**

> (i)
> 1. The above messages can be logged with an A or B suffix for an ASCII or binary output with a NovAtel header followed by Hex or binary raw data respectively.
> 2. When you plan to send both RTCAOBS2 and RTCAOBS messages, ensure you send the RTCAOBS2 message before RTCAOBS.

The RTCA (Radio Technical Commission for Aviation Services) Standard is being designed to support Differential Global Navigation Satellite System (DGNSS) Special Category I (SCAT-I) precision instrument approaches. The RTCA Standard is in a preliminary state. Described below is NovAtel's current support for this standard. It is based on "Minimum Aviation System Performance Standards DGNSS Instrument Approach System: Special Category I (SCAT-I)".[1]

NovAtel has defined six proprietary RTCA Standard Type 7 binary format messages, RTCAOBS, RTCAOBS2, RTCAREF, RTCAEPHEM, RTCAREFEXT and RTCAOBS3 for base station transmissions. RTCAOBS3 and RTCAREFEXT are defined specifically for use in ALIGN. These can be used with either single- or dual-frequency NovAtel receivers. The RTCA message format out performs the RTCM format in the following ways, among others:

- a more efficient data structure (lower overhead)
- better error detection
- allowance for a longer message, if necessary

1.For further information about RTCA Standard messages, refer to:
   Minimum Aviation System Performance Standards - DGNSS Instrument Approach System: Special Category I (SCAT-I), Document No. RTCA/DO-217 (April 19,1995); Appx A, Pg 21

RTCAREF and RTCAOBS, respectively, correspond to the RTCM Type 3 and Type 59 logs used in single-frequency only measurements. Both are NovAtel proprietary RTCA Standard Type 7 messages with an 'N' primary sub-label.

Refer to the *Receiving and Transmitting Corrections* section in the OEM6 Family Installation and Operation User Manual (OM-20000128) for more information about using these message formats for differential operation.

Input Example

```
INTERFACEMODE COM2 NONE RTCA
FIX POSITION 51.1136 -114.0435 1059.4
LOG COM2 RTCAOBS2 ONTIME 1
LOG COM2 RTCAOBS ONTIME 1
LOG COM2 RTCAREF ONTIME 10
LOG COM2 RTCA1 ONTIME 5
LOG COM2 RTCAEPHEM ONTIME 10 1
LOG COM2 RTCAREFEXT ONTIME 1
LOG COM2 RTCAOBS3 ONTIME 1
```

3.2.129 RTCM Standard Logs

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

RTCM1	**Differential GPS Corrections**	
Message ID:	**107**	
RTCM3	**Base Station parameters**	
Message ID:	**117**	
RTCM9	**Partial Differential GPS Corrections**	
Message ID:	**275**	
RTCM15	**Ionospheric Corrections**	
Message ID:	**307**	
RTCM16	**Special Message**	
Message ID:	**129**	
RTCM16T	**SPECIAL Text MESSAGE,** see also *page 302*	
Message ID:	**131**	
RTCM1819	**Raw Measurements**	
Message ID:	**260**	
RTCM2021	**Measurement Corrections**	
Message ID:	**374**	
RTCM22	**Extended Base Station**	
Message ID:	**118**	
RTCM23	**Antenna Type Definition**	
Message ID:	**665**	
RTCM24	**Antenna Reference Point (ARP)**	
Message ID:	**667**	
RTCM31	**Differential GLONASS**	
Message ID:	**864**	
RTCM32	**GLONASS Base parameters**	
Message ID:	**873**	
RTCM36	**SPECIAL Extended MESSAGE**	
Message ID:	**875**	
RTCM36T	**SPECIAL EXTENDED MESSAGE,** see also *page 303*	
Message ID:	**877**	
RTCM59	**Type 59N-0 Proprietary Differential**	
Message ID:	**116**	
RTCM59GLO	**Proprietary GLONASS Differential**	
Message ID:	**903**	
RTCMOMNI1	**RTCM1 from OmniSTAR VBS**	
Message ID:	**957**	

(i) 1. The RTCM messages can be logged with an A or B suffix for an ASCII or binary output with a NovAtel header followed by Hex or binary raw data respectively.

2. Combinations of integer offsets and fractional offsets are not supported for RTCM logs. See also the LOG command starting on *page 202* for more details on offsets.

The Radio Technical Commission for Maritime Services (RTCM) was established to facilitate the establishment of various radio navigation standards, which includes recommended GNSS differential standard formats. Refer to the *Receiving and Transmitting Corrections* section in the [OEM6 Family Installation and Operation User Manual](#) (OM-200000128) for more information about using these message formats for differential operation.

The standards recommended by the RTCM Special Committee 104, Differential GPS Service (RTCM SC-104,Washington, D.C.), have been adopted by NovAtel for implementation into the receiver. Because the receiver is capable of utilizing RTCM formats, it can easily be integrated into positioning systems around the globe.

As it is beyond the scope of this manual to provide indepth descriptions of the RTCM data formats, it is recommended that anyone requiring explicit descriptions should obtain a copy of the published RTCM specifications.

RTCM SC-104[1] Type 3 and 59 messages can be used for base station transmissions in differential systems. However, since these messages do not include information about the L2 component of the GPS signal, they cannot be used with RT-2 positioning. Regardless of whether single or dual-frequency receivers are used, the RT-20 positioning algorithm is used. This is for a system in which both the base and rover stations utilize NovAtel receivers.

Note that the error detection capability of an RTCM format message is less than that of an RTCA-format message. The communications equipment used may have an error detection capability of its own to supplement the RTCM message which induces higher overhead. Consult the radio vendor's documentation for further information.

If RTCM format messaging is being used, the optional station id field that is entered using the DGPSTXID command (*page 125*)can be any number within the range of 0 - 1023 (for example, 119). The representation in the log message is identical to what was entered.

The NovAtel logs which implement the RTCM Standard Format for Type 1, 3, 9, 16, 18, 19, 22, 23, 24, 31, 32 and 36 messages are known as the RTCM1, RTCM3, RTCM9, RTCM16, RTCM1819, RTCM22, RTCM23, RTCM24, RTCM31, RTCM32 and RTCM36 logs, respectively, while Type 59N-0 messages are listed in the RTCM59 log.

---

1. For further information about RTCM SC-104 messages, you may wish to refer to: RTCM Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service, Version 2.3 at http://www.rtcm.org/overview.php.

All receiver RTCM standard format logs adhere to the structure recommended by RTCM SC-104. Thus, all RTCM message are composed of 30 bit words. Each word contains 24 data bits and 6 parity bits. All RTCM messages contain a 2 word header followed by 0 to 31 data words for a maximum of 33 words (990 bits) per message.

Message Frame Header	Data	Bits
Word 1	Message frame preamble for synchronization	8
	Frame/message type ID	6
	Base station ID	10
	Parity	6
Word 2	Modified z-count (time tag)	13
	Sequence number	3
	Length of message frame	5
	Base health	3
	Parity	6

Version 3.0, also developed by the RTCM SC-104, consists primarily of messages designed to support Real-Time Kinematic (RTK) operations. It provides messages that support GPS and GLONASS RTK operations, including code and carrier phase observables, antenna parameters, and ancillary system parameters. Version 3.1 adds RTCM messages containing transformation data and information about Coordinate Reference Systems.[1]

The remainder of this section provides further information concerning receiver commands and logs that utilize the RTCM data formats.

Example Input:
```
INTERFACEMODE COM2 NONE RTCM
FIX POSITION 51.1136 -114.0435 1059.4
LOG COM2 RTCM3 ONTIME 10
LOG COM2 RTCM22 ONTIME 10 1
LOG COM2 RTCM1819 ONTIME 1
LOG COM2 RTCM31 ONTIME 2
LOG COM2 RTCM32 ONTIME 2
LOG COM2 RTCM1 ONTIME 5
```

---

1.   For further information about RTCM SC-104 messages, refer to:
      RTCM Recommended Standards for Differential GNSS (Global Navigation
      Satellite Systems) Service, Version 3.0 and Version 3.1 at http://www.rtcm.org/
      overview.php.

OmniSTAR Local Wide Area Corrections

RTCM Type 1 messages are generated from OmniSTAR Virtual Base Station (VBS) corrections.

The positioning performance using OmniSTAR local wide area corrections meets the standard OmniSTAR VBS code differential performance specifications.

Unless otherwise noted, values in the RTCM Type 1 messages are unchanged from what is provided by the VBS library (for example, RRC, UDRE, station ID) apart from necessary unit scaling. An RTCM1 message is generated and output each time the VBS library provides updated corrections (about every 6 s). The receiver no longer outputs corrections when the L-Band signal is lost and the VBS library stops generating corrections. The output is for the same set of satellites provided by the VBS library (above 5° elevation at the current position).

Enable the output of OmniSTAR VBS corrections in RTCM messages by using the following commands:

```
INTERFACEMODE COM2 NOVATEL RTCM OFF
ASSIGNLBAND OMNISTAR <FREQUENCY> <BPS> OR ASSIGNLBAND OMNISTARAUTO
PSRDIFFSOURCE OMNISTAR
LOG COM2 RTCMOMNI1 ONCHANGED
```

The RTCMOMNI1 log is asynchronous.

The OmniSTAR RTCM model outputs RTCM corrections at a rate of up to 0.2 Hz. This new model does not include position or raw measurement output.

3.2.130 RTCMV3 Standard Logs

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

RTCM1001	**L1-Only GPS RTK Observables**
Message ID:	**772**

RTCM1002	**Extended L1-Only GPS RTK Observables**
Message ID:	**774**

RTCM1003	**L1 And L2 GPS RTK Observables**
Message ID:	**776**

RTCM1004	**Extended L1 and L2 GPS RTK Observables**
Message ID:	**770**

RTCM1005	**Stationary RTK Base Station Antenna Reference Point (ARP)**
Message ID:	**765**

RTCM1006	**Stationary RTK Base Station ARP with Antenna Height**
Message ID:	**768**

RTCM1007	**Extended Antenna Descriptor and Setup Information**
Message ID:	**852**

RTCM1008	**Extended Antenna Reference Station Description and Serial Number**
Message ID:	**854**

RTCM1009	**GLONASS L1-Only RTK**
Message ID:	**885**

RTCM1010	**Extended GLONASS L1-Only RTK**
Message ID:	**887**

RTCM1011	**GLONASS L1/L2 RTK**
Message ID:	**889**

RTCM1012	**Extended GLONASS L1/L2 RTK**
Message ID:	**891**

RTCM1019	**GPS Ephemerides**
Message ID:	**893**

RTCM1020	**GLONASS EPHEMERIDES**
Message ID:	**895**

RTCM1033	**Receiver and antenna descriptors**
Message ID:	**1097**

RTCM1071	**MSM1, GPS Code Measurements**
Message ID:	**1472**

RTCM1072	**MSM2, GPS Phase Measurements**
Message ID:	**1473**

RTCM1073	**MSM3, GPS Code and Phase Measurements**
Message ID:	**1474**

RTCM1074	**MSM4, GPS Code, Phase and CNR Measurements**
Message ID:	**1475**

RTCM1075	**MSM5, GPS Code, Phase, CNR and Doppler Measurements**
Message ID:	**1476**

RTCM1076	**MSM6, Extended GPS Code, Phase and CNR Measurements**
Message ID:	**1477**

RTCM1077	**MSM7, Extended GPS Code, Phase, CNR and Doppler Measurements**
Message ID:	**1478**
RTCM1081	**MSM1, GLONASS Code Measurements**
Message ID:	**1479**
RTCM1082	**MSM2, GLONASS Phase Measurements**
Message ID:	**1480**
RTCM1083	**MSM3, GLONASS Code and Phase Measurements**
Message ID:	**1481**
RTCM1084	**MSM4, GLONASS Code, Phase and CNR Measurements**
Message ID:	**1482**
RTCM1085	**MSM5, GLONASS Code, Phase, CNR and Doppler Measurements**
Message ID:	**1483**
RTCM1086	**MSM6, Extended GLONASS Code, Phase and CNR Measurements**
Message ID:	**1484**
RTCM1087	**MSM7, Extended GLONASS Code, Phase, CNR and Doppler Measurements**
Message ID:	**1485**
RTCM1091	**MSM1, Galileo Code Measurements**
Message ID:	**1486**
RTCM1092	**MSM2, Galileo Phase Measurements**
Message ID:	**1487**
RTCM1093	**MSM3, Galileo Code and Phase Measurements**
Message ID:	**1488**
RTCM1094	**MSM4, Galileo Code, Phase and CNR Measurements**
Message ID:	**1489**
RTCM1095	**MSM5, Galileo Code, Phase, CNR and Doppler Measurements**
Message ID:	**1490**
RTCM1096	**MSM6, Extended Galileo Code, Phase and CNR Measurements**
Message ID:	**1491**
RTCM1097	**MSM7, Extended Galileo Code, Phase, CNR and Doppler Measurements**
Message ID:	**1492**
RTCM1111	**MSM1, QZSS Code Measurements**
Message ID:	**1648**
RTCM1112	**MSM2, QZSS Phase Measurements**
Message ID:	**1649**
RTCM1113	**MSM3, QZSS Code and Phase Measurements**
Message ID:	**1650**
RTCM1114	**MSM4, QZSS Code, Phase and CNR Measurements**
Message ID:	**1651**
RTCM1115	**MSM5, QZSS Code, Phase, CNR and Doppler Measurements**
Message ID:	**1652**
RTCM1116	**MSM6, Extended QZSS Code, Phase and CNR Measurements**
Message ID:	**1653**
RTCM1117	**MSM7, Extended QZSS Code, Phase, CNR and Doppler Measurements**
Message ID:	**1654**

RTCM1121   **MSM1, BeiDou Code Measurements**
Message ID: **1592**

RTCM1122   **MSM2, BeiDou Phase Measurements**
Message ID: **1593**

RTCM1123   **MSM3, BeiDou Code and Phase Measurements**
Message ID: **1594**

RTCM1124   **MSM4, BeiDou Code, Phase and CNR Measurements**
Message ID: **1595**

RTCM1125   **MSM5, BeiDou Code, Phase, CNR and Doppler Measurements**
Message ID: **1596**

RTCM1126   **MSM6, Extended BeiDou Code, Phase and CNR Measurements**
Message ID: **1597**

RTCM1127   **MSM7, Extended BeiDou Code, Phase, CNR and Doppler Measurements**
Message ID: **1598**

(i) 1. At the base station, choose to send either an RTCM1005 or RTCM1006 message to the rover station. Then select one of the observable messages (RTCM1001, RTCM1002, RTCM1003 or RTCM1004) to send from the base.

2. RTCM1007 and RTCM1008 data is set using the `BASEANTENNAMODEL` command, see *page 84*. If you have set a base station ID, it is detected and set. Other values are also taken from a previously entered `BASEANTENNAMODEL` command.

3. In order to set up logging of RTCM1007 or RTCM1008 data, it is recommended to first use the INTERFACEMODE command to set the interface mode of the port transmitting RTCMV3 messages to RTCMV3, see *page 184*. Providing the base has a fixed position, see FIX on *page 154* or is configured as a moving base station (refer to the `MOVINGBASESTATION` command on *page 217*) and its `BASEANTENNAMODEL` command set, you can log out RTCM1007 messages.

4. The RTCM messages can be logged with an A or B suffix for an ASCII or binary output with a NovAtel header followed by Hex or binary raw data respectively.

RTCM SC-104 is a more efficient alternative to the documents entitled "RTCM Recommended Standards for Differential NAVSTAR GPS Service, Version 2.x". Version 3.0, consists primarily of messages designed to support RTK operations. The reason for this emphasis is that RTK operation involves broadcasting a lot of information and thus benefits the most from a more efficient data format.

The RTCM SC-104 standards have been adopted by NovAtel for implementation into the receiver. The receiver can easily be integrated into positioning systems around the globe because it is capable of utilizing RTCM Version 3.0 formats.

The initial Version 3.0 document describes messages and techniques for supporting GPS. The format accommodates modifications to these systems (for example, new signals) and to new satellite systems that are under development. In addition, augmentation systems that utilize geostationary satellites, with transponders operating in the same frequency bands, are now in the implementation stages. Generically, they are called Satellite-Based Augmentation Systems (SBAS) and are designed to be interoperable (for example WAAS, EGNOS, MSAS).

Message types contained in the current Version 3.0 standard have been structured in different groups. Transmit at least one message type from each of Groups 1 to 3:

Group 1 - Observations:

RTCM1001	L1-Only GPS RTK
RTCM1002	Extended L1 Only GPS RTK
RTCM1003	L1 And L2 GPS RTK
RTCM1004	Extended L1and L2 GPS RTK
RTCM1009	L1-Only GLONASS RTK
RTCM1010	Extended L1 Only GLONASS RTK
RTCM1011	L1/L2 GLONASS RTK
RTCM1012	Extended L1/L2 GLONASS RTK

Group 2 - Base Station Coordinates:

RTCM1005	RTK Base Antenna Reference Point (ARP)
RTCM1006	RTK Base ARP with Antenna Height

Group 3 - Antenna Description:

RTCM1007	Extended Antenna Descriptor and Setup Information
RTCM1008	Extended Antenna Reference Station Description and Serial Number

Group 4 - Auxiliary Operation Information:

RTCM1019	GPS Ephemerides
RTCM1020	GLONASS Ephemerides
RTCM1033	Receiver and Antenna Descriptors

Example Input:

```
INTERFACEMODE COM2 NONE RTCMV3
FIX POSITION 51.1136 -114.0435 1059.4
BASEANTENNAMODEL 702 NVH05410007 1 USER
LOG COM2 RTCM1005 ONTIME 10
LOG COM2 RTCM1002 ONTIME 5
LOG COM2 RTCM1007 ONTIME 10
```

RTCM1001-RTCM1004 GPS RTK Observables

RTCM1001, RTCM1002, RTCM1003 and RTCM1004 are GPS RTK messages, which are based on raw data. From this data, valid RINEX files can be obtained. As a result, this set of messages offers a high level of interoperability and compatibility with standard surveying practices. Refer also to the *NovAtel PC Utilities* manual on the CD with your product for details on the logs that Convert4 converts to RINEX.

The Type 1001 Message supports single-frequency RTK operation. It does not include an indication of the satellite Carrier-to-Noise (C/No) as measured by the base station.

The Type 1002 Message supports single-frequency RTK operation and includes an indication of the satellite C/No as measured by the base station. Since the C/No does not usually change from measurement to measurement, this message type can be mixed with the Type 1001 and is used primarily when a satellite C/No changes, thus saving broadcast link throughput.

The Type 1003 Message supports dual-frequency RTK operation, but does not include an indication of the satellite C/No as measured by the base station.

The Type 1004 Message supports dual-frequency RTK operation, and includes an indication of the satellite C/No as measured by the base station. Since the C/No does not usually change from measurement to measurement, this message type can be mixed with the Type 1003 and is used only when a satellite C/No changes, thus saving broadcast link throughput.

RTCM1005 and RTCM1006 RTK Base Antenna Reference Point (ARP)

Message Type 1005 provides the Earth-Centered, Earth-Fixed (ECEF) coordinates of the ARP for a stationary base station. No antenna height is provided.

Message Type 1006 provides all the same information as Message Type 1005 and also provides the height of the ARP.

These messages are designed for GPS operation and are equally applicable to future satellite systems. System identification bits are reserved for them.

Message Types 1005 and 1006 avoid any phase center problems by utilizing the ARP, which is used throughout the International GPS Service (IGS). They contain the coordinates of the installed antenna's ARP in ECEF coordinates; datum definitions are not yet supported. The coordinates always refer to a physical point on the antenna, typically the bottom of the antenna mounting surface.

RTCM1007 and RTCM1008 Extended Antenna Descriptions

Message Type 1007 provides an ASCII descriptor of the base station antenna. The International GPS Service (IGS) Central Bureau convention is used most of the time, since it is universally accessible.

Message Type 1008 provides the same information, plus the antenna serial number, which removes any ambiguity about the model number or production run.

IGS limits the number of characters to 20. The antenna setup ID is a parameter for use by the service provider to indicate the particular base station-antenna combination. "0" for this value means that the values of a standard model type calibration should be used. The antenna serial number is the individual antenna serial number as issued by the manufacturer of the antenna.

RTCM1009-RTCM1012 GLONASS RTK Observables

Message Types 1009 through 1012 provide the contents of the GLONASS RTK messages, which are based on raw data. You can obtain complete RINEX files from this data. This set of messages offers a high level of interoperability and compatibility with standard surveying practices. When using these messages, you should also use an ARP message (Type 1005 or 1006) and an Antenna Descriptor message (Type 1007 or 1008). If the time tags of the GPS and GLONASS RTK data are synchronized, the Synchronized GNSS flag can be used to connect the entire RTK data block.

RTCM1019-RTCM1020 GPS and GLONASS Ephemerides

Message Type 1019 contains GPS satellite ephemeris information. Message Type 1020 contains GLONASS ephemeris information. These messages can be broadcast in the event that an anomaly in ephemeris data is detected, requiring the base station to use corrections from previously good satellite ephemeris data. This allows user equipment just entering the differential system to use corrections broadcast from that ephemeris. Broadcast this message (Type 1019 or 1020) every 2 minutes until the satellite broadcast is corrected or until the satellite drops below the coverage area of the base station.

These messages can also be used to assist receivers to quickly acquire satellites. For example, if you access a wireless service with this message, it can utilize the ephemeris information immediately rather than waiting for a satellite to be acquired and the almanac data processed.

RTCM1070-RTCM1229 Multiple Signal Messages (MSM)

The MSM messages are a set of RTK correction messages that provide standardized content across all current and future GNSS system.

Each GNSS system has a set of seven MSM types numbered from 1 to 7. The MSM type for each GNSS system provides the same generic information. For example, MSM1 for each GNSS system provides the code measurements for the system. See *Table 136, MSM type descriptions* on page 646 for the descriptions of each of the seven MSM types.

Table 136: MSM type descriptions

Message	Description
MSM1	Provides the code measurements.
MSM2	Provides the phase measurements.
MSM3	Provides the data from MSM1 (code) and MSM2 (phase) in a single message.
MSM4	Provides all the data from MSM3 (code and phase) and adds the CNR measurements.
MSM5	Provides all the data from MSM4 (code, phase and CNR) and adds the doppler measurements.
MSM6	Provides the same information as MSM4, but has extended resolution on the measurements.
MSM7	Provides the same information as MSM5, but has extended resolution on the measurements.

Table 137, Supported MSM messages lists the MSM messages supported on OEM6.

Table 137: Supported MSM messages

Message	GPS	GLONASS	Galileo	QZSS	BeiDou
MSM1	RTCM1071	RTCM1081	RTCM1091	RTCM1111	RTCM1121
MSM2	RTCM1072	RTCM1082	RTCM1092	RTCM1112	RTCM1122
MSM3	RTCM1073	RTCM1083	RTCM1093	RTCM1113	RTCM1123
MSM4	RTCM1074	RTCM1084	RTCM1094	RTCM1114	RTCM1124
MSM5	RTCM1075	RTCM1085	RTCM1095	RTCM1115	RTCM1125
MSM6	RTCM1076	RTCM1086	RTCM1096	RTCM1116	RTCM1126
MSM7	RTCM1077	RTCM1087	RTCM1097	RTCM1117	RTCM1127

For most applications, MSM3 is recommended.

3.2.131 RTKDOP
DOP values from the RTK fast filter

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the DOP values calculated by the RTK fast filter.

The RTKDOP log contains single-point DOPs, calculated using only the satellites used in the fast RTK solution, that is, those used for the RTKPOS position. Calculation of the RTK DOPs are limited to once a second.

The calculation of the RTK DOP is different than that for the pseudorange DOP. In the pseudorange filter, new DOPs are calculated every 60 seconds. The RTK DOP is calculated at the rate requested and regardless of a change in satellites. However, the DOP is only calculated when the RTKDOP log is requested.

Message ID: **952**
Log Type: **Synch**

Recommended Input:

       log rtkdopa ontime 10

```
ASCII EXAMPLE:
#RTKDOPA,COM1,0,60.0,FINESTEERING,1449,446982.000,00000008,B42B,
3044;2.3386,1.9856,0.9407,1.5528,1.2355,10.0,11,21,58,6,7,10,16,18,
24,26,29,41*85F8338B
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RTKDOP header	Log header		H	0
2	GDOP	Geometric DOP	Float	4	H
3	PDOP	Position DOP	Float	4	H+4
4	HDOP	Horizontal DOP	Float	4	H+8
5	HTDOP	Horizontal and Time DOP	Float	4	H+12
6	TDOP	Time DOP	Float	4	H+16
7	elev mask	GPS elevation mask angle	Float	4	H+20
8	#sats	Number of satellites to follow	Ulong	4	H+24
9	sats	Satellites in use at time of calculation	Ulong	4	H+28
10	Next satellite offset = H+28+(#sats * 4)				
11	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+28+ (#sats * 4)
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.132 RTKDOP2
DOP values from the RTK low latency filter

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log is similar to the RTKDOP log, but contains the per-system TDOPs.

Message ID: **1172**
Log Type: **Synch**

Recommended Input:

> log rtkdop2a ontime 10

ASCII Example:

```
#RTKDOP2A,COM1,0,80.0,FINESTEERING,1690,601478.000,00000008,AB50,
43488;1.5000,1.1850,0.6580,0.9850,2,GPS,0.6530,GLONASS,0.6490
*C5F1A25F
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RTKDOP2 header	Log header		H	0
2	GDOP	Geometric DOP	Float	4	H
3	PDOP	Position DOP	Float	4	H+4
4	HDOP	Horizontal DOP	Float	4	H+8
5	VDOP	Vertical DOP	Float	4	H+12
6	#systems	Number of entries to follow	ULong	4	H+16
7	system	See *Table 64, System Used for Timing* on *page 308*	Enum	4	H+20
8	TDOP	Time DOP (Dilution of Precision)	Float	4	H+24
9	Next satellite offset = H+20+(#systems * 8)				
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+20+ (#systems * 8)
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.133 RTKPOS
RTK low latency position data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the low latency RTK position computed by the receiver, along with two status flags. In addition, it reports other status indicators, including differential age, which is useful in predicting anomalous behavior brought about by outages in differential corrections. This log is recommended for kinematic operation. Better accuracy can be obtained in static operation with the MATCHEDPOS log (*page 540*).

With the system operating in an RTK mode, this log reflects if the solution is a good RTK low latency solution (from extrapolated base station measurements) or invalid. A valid RTK low latency solution is computed for up to 60 seconds after reception of the last base station observation. The degradation in accuracy, due to differential age, is reflected in the standard deviation fields, and is summarized in the [Standards and References](#) section of our website [www.novatel.com/support/](#). See also the PSRDIFFTIMEOUT command (*page 250*).

Message ID: **141**
Log Type: **Synch**

Recommended Input:

> log rtkposa ontime 1

```
ASCII EXAMPLE:
#RTKPOSA,COM1,0,54.5,FINESTEERING,1419,340040.000,00000040,176E,
2724;SOL_COMPUTED,NARROW_INT,51.11635911294,-114.03833103654,
1063.8336,-16.2712,WGS84,0.0179,0.0096,0.0174,"AAAA",1.000,0.000,
12,11,11,11,0,01,0,33*0ADB3E47
```

> (i) Consider the case of a racing car, on a closed circuit, requiring RTK operation. In this situation, you would have to send live data to the pits using a radio link.
>
> RTK operation enables live centimeter level position accuracy. When answers are required in the field, the base station must transmit information to the rover in real-time. For RTK operation, extra equipment such as radios are required to transmit and receive this information. The base station has a corresponding base radio and the rover station has a corresponding rover radio.
>
> Post-processing can provide post-mission position and velocity data using raw GNSS data collected from the car. The logs necessary for post-processing include:
>
> > RANGECMPB ONTIME 1
> > RAWEPHEMB ONNEW
>
> These are examples of data collection for post-processing, and real-time operation. OEM6-based output is compatible with post-processing software from the NovAtel's Waypoint Products Group or refer to our website at [www.novatel.com.](#). for more details.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RTKPOS header	Log header		H	0
2	sol status	Solution status (see *Table 85* on *page 408*)	Enum	4	H
3	pos type	Position type (see *Table 84* on *page 407*)	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 (m)	Double	8	H+24
7	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid (m) [a]	Float	4	H+32
8	datum id#	Datum ID number (see *Table 23, Reference Ellipsoid Constants* on *page 119*)	Enum	4	H+36
9	lat σ	Latitude standard deviation (m)	Float	4	H+40
10	lon σ	Longitude standard deviation (m)	Float	4	H+44
11	hgt σ	Height standard deviation (m)	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 satellites tracked	Uchar	1	H+64
16	#solnSVs	Number of satellites vehicles used in solution	Uchar	1	H+65
17	#ggL1	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		Hex	1	H+68
20	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+69
21	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+70
22	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	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.

3.2.134 RTKSATS
Satellites used in RTKPOS solution

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log lists the used and unused satellites for the corresponding RTKPOS solution. It also describes the signals of the used satellites and reasons for exclusions.

Message ID:	**1174**
Log Type:	**Synch**

Recommended Input:

> log rtksats ontime 1

Abbreviated ASCII Example:

```
<RTKSATS COM1 0 60.5 FINESTEERING 1728 524924.000 00000000 95E7 11487
<     24
<          GPS 3 GOOD 00000003
<          GPS 5 GOOD 00000003
...
<          GPS 23 GOOD 00000003
<          GPS 30 GOOD 00000003
<          GLONASS 1+1 GOOD 00000003
<          GLONASS 2-4 GOOD 00000003
...
<          GLONASS 20+2 GOOD 00000003
<          GLONASS 21+4 GOOD 00000003
<          BEIDOU 6 GOOD 00000003
<          BEIDOU 11 GOOD 00000003
...
<          BEIDOU 12 GOOD 00000003
<          BEIDOU 13 GOOD 00000003
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RTKSATS header	Log header		H	0
2	#entries	Number of records to follow	Ulong	4	H
3	system	Satellite system (refer to *Table 110* on *page 508*)	Enum	4	H+4
4	Satellite ID	Satellite identifiers	Ulong	4	H+8
5	Status	Satellite status. See *Table 90* on *page 413*	Enum	4	H+12
6	Signal mask	See *Table 91* through *Table 94*	Hex	4	H+16
7	Next satellite offset = H+4+(#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.135 RTKVEL
RTK velocity

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the RTK velocity information computed by the receiver. In addition, it reports a velocity status indicator that is useful in indicating whether or not the corresponding data is valid and differential age is useful in predicting anomalous behavior brought about by outages in differential corrections. 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. See also the table footnote for velocity logs on page 362.

> (i) Velocities from the RTK filter are calculated from the delta-position. In RTKVEL, the velocity type is the same as the position type.

With the system operating in an RTK mode, this log reflects if the solution is a good RTK low latency solution (from extrapolated base station measurements) or invalid. A valid RTK low latency solution is computed for up to 60 seconds after reception of the last base station observation.

The velocity is computed from consecutive RTK low latency updates. As such, it is an average velocity based on the time difference between successive position computations and not an instantaneous velocity at the RTKVEL time tag. The velocity latency to be subtracted from the time tag is normally half the time between filter updates. Under default operation, the RTK low latency filter is updated at a rate of 2 Hz. This translates into a velocity latency of 0.25 seconds. The latency can be reduced by increasing the update rate of the RTK low latency filter by requesting the BESTVEL, RTKVEL, BESTPOS or RTKPOS messages at a rate higher than 2 Hz. For example, a logging rate of 10 Hz would reduce the velocity latency to 0.05 seconds. For integration purposes, the velocity latency should be applied to the record time tag.

Message ID: **216**
Log Type: **Synch**

Recommended Input:

>   log rtkvela ontime 1

ASCII Example:

```
#RTKVELA,COM1,0,43.5,FINESTEERING,1364,496137.000,00100000,71E2,
2310;SOL_COMPUTED,NAR-
ROW_INT,0.250,1.000,0.0027,207.645811,0.0104,0.0*F551CC42
```

> (i) Consider the case of an unmanned aircraft. A base station must send differential correction data to the remote aircraft. In this type of application, the aircraft's radio may pass the differential solution, for example RTKVEL, to the positioning system so it can process it and generate precise position information for the flight controls.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RTKVEL header	Log header		H	0
2	sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	vel type	Velocity type, see *Table 84, Position or Velocity Type* on *page 407*	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 in seconds	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)	Ulong	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.136 RTKXYZ
RTK Cartesian position and velocity

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the receiver's low latency position and velocity in ECEF coordinates. The position and velocity status fields indicate whether or not the corresponding data is valid. See *Figure 11, The WGS84 ECEF Coordinate System* on *page 422* for a definition of the ECEF coordinates.

The velocity measurements have a latency associated with them. The time of validity is the time tag in the log minus the latency value.

With the system operating in an RTK mode, this log reflects if the solution is a good RTK low latency solution (from extrapolated base station measurements) or invalid. A valid RTK low latency solution is computed for up to 60 seconds after reception of the last base station observation. The degradation in accuracy due to differential age is reflected in the standard deviation fields, and is summarized in the Standards and References section of our website www.novatel.com/support/. See also the PSRDIFFTIMEOUT command (*page 250*).

The velocity is computed from consecutive RTK low latency updates. As such, it is an average velocity based on the time difference between successive position computations and not an instantaneous velocity at the RTKVEL time tag. The velocity latency to be subtracted from the time tag is normally half the time between filter updates. Under default operation, the RTK low latency filter is updated at a rate of 2 Hz. This translates into a velocity latency of 0.25 seconds. The latency can be reduced by increasing the update rate of the RTK low latency filter by requesting the BESTXYZ message at a rate higher than 2 Hz. For example, a logging rate of 10 Hz would reduce the velocity latency to 0.05 seconds. For integration purposes, the velocity latency should be applied to the record time tag.

See also the BESTXYZ and MATCHEDXYZ logs, on *pages 420* and *543*.

Message ID: **244**
Log Type: **Synch**

Recommended Input:

        log rtkxyza ontime 1

ASCII Example:

```
#RTKXYZA,COM1,0,56.0,FINESTEERING,1419,340041.000,00000040,3D88,
2724;SOL_COMPUTED,NARROW_INT,-1634531.5666,-3664618.0291,
4942496.3230,0.0099,0.0219,0.0115,SOL_COMPUTED,NARROW_INT,0.0030,
0.0003,-0.0016,0.0198,0.0438,0.0230,"AAAA",0.250,1.000,0.000,12,11,
11,11,0,01,0,33*0497D146
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RTKXYZ header	Log header		H	0
2	P-sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H
3	pos type	Position type, see *Table 84, Position or Velocity Type* on *page 407*	Enum	4	H+4
4	P-X	Position X-coordinate (m)	Double	8	H+8
5	P-Y	Position Y-coordinate (m)	Double	8	H+16
6	P-Z	Position Z-coordinate (m)	Double	8	H+24
7	P-X σ	Standard deviation of P-X (m)	Float	4	H+32
8	P-Y σ	Standard deviation of P-Y (m)	Float	4	H+36
9	P-Z σ	Standard deviation of P-Z (m)	Float	4	H+40
10	V-sol status	Solution status, see *Table 85, Solution Status* on *page 408*	Enum	4	H+44
11	vel type	Velocity type, see *Table 84* on *page 407*	Enum	4	H+48
12	V-X	Velocity vector along X-axis (m)	Double	8	H+52
13	V-Y	Velocity vector along Y-axis (m)	Double	8	H+60
14	V-Z	Velocity vector along Z-axis (m)	Double	8	H+68
15	V-X σ	Standard deviation of V-X (m)	Float	4	H+76
16	V-Y σ	Standard deviation of V-Y (m)	Float	4	H+80
17	V-Z σ	Standard deviation of V-Z (m)	Float	4	H+84
18	stn ID	Base station identification	Char[4]	4	H+88
19	V-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+92
20	diff_age	Differential age in seconds	Float	4	H+96
21	sol_age	Solution age in seconds	Float	4	H+100
22	#SVs	Number of satellites tracked	Uchar	1	H+104
23	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+105
24	#ggL1	Number of satellites with L1/E1/B1 signals used in solution	Uchar	1	H+106
25	#solnMultiSVs	Number of satellites with multi-frequency signals used in solution	Uchar	1	H+107
26	Reserved		Char	1	H+108
27	ext sol stat	Extended solution status (see *Table 88, Extended Solution Status* on *page 410*)	Hex	1	H+109

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
28	Galileo and BeiDou sig mask	Galileo and BeiDou signals used mask (see *Table 87, BESTPOS Galileo and BeiDou Signal-Used Mask* on *page 409*)	Hex	1	H+110
29	GPS and GLONASS sig mask	GPS and GLONASS signals used mask (see *Table 86, BESTPOS GPS and GLONASS Signal-Used Mask* on *page 409*)	Hex	1	H+111
30	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+112
31	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.137 RXCONFIG
Receiver configuration

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log is used to output a list of all current command settings. When requested, an RXCONFIG log is output for each setting. See also the LOGLIST log on *page 526* for a list of currently active logs. One log is output for each current command.

Message ID: **128**
Log Type: **Polled**

Recommended Input:
```
log rxconfiga once
```

ASCII Example[1]:
```
#RXCONFIGA,COM1,71,47.5,APPROXIMATE,1337,333963.260,00000000,F702,
1984;
#ADJUST1PPSA,COM1,71,47.5,APPROXIMATE,1337,333963.260,00000000,
F702,1984;OFF,ONCE,0*BA85A20B*91F89B07
#RXCONFIGA,COM1,70,47.5,APPROXIMATE,1337,333963.398,00000000,F702,
1984;
#ANTENNAPOWERA,COM1,70,47.5,APPROXIMATE,1337,333963.398,00000000,
F702,1984;ON*D12F6135*8F8741BE
#RXCONFIGA,COM1,69,47.5,APPROXIMATE,1337,333963.455,00000000,F702,
1984;
#CLOCKADJUSTA,COM1,69,47.5,APPROXIMATE,1337,333963.455,00000000,
F702,1984;ENABLE*0AF36D92*B13280F2
...
#RXCONFIGA,COM1,7,47.5,APPROXIMATE,1337,333966.781,00000000,F702,
1984;
#STATUSCONFIGA,COM1,7,47.5,APPROXIMATE,1337,333966.781,00000000,
F702,1984;CLEAR,AUX2,0*A6141E28*D0BBA9F2
#RXCONFIGA,COM1,2,47.5,APPROXIMATE,1337,333967.002,00000000,F702,
1984;
#SBASECUTOFFA,COM1,2,47.5,APPROXIMATE,1337,333967.002,00000000,
F702,1984;-5.000000000*B9B11096*2E8B77CF
#RXCONFIGA,COM1,1,47.5,FINESTEERING,1337,398382.787,00000000,F702,
1984;
#LOGA,COM1,1,47.5,FINESTEERING,1337,398382.787,00000000,F702,1984;
COM1,MARKPOSA,ONNEW,0.000000,0.000000,NOHOLD*A739272D*6692C084
#RXCONFIGA,COM1,0,47.5,FINESTEERING,1337,400416.370,00000000,F702,
1984;
#LOGA,COM1,0,47.5,FINESTEERING,1337,400416.370,00000000,F702,1984;
COM2,PASSCOM2A,ONCHANGED,0.000000,0.000000,NOHOLD*55FC0C62*17086D18
```

---

1. The embedded CRCs are flipped to make the embedded messages recognizable to the receiver. For example, consider the first embedded message above.

   91f89b07:   100100011111100010011011000000111
                      111000001101100100001111110001001:e0d91f89

   The CRC is really e0d91f89.

> Do not use undocumented commands or logs! Doing so may produce errors and void your warranty.

> The RXCONFIG log can be used to ensure your receiver is correctly setup for your application.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RXCONFIG header	Log header	-	H	0
2	e header	Embedded header	-	h	H
3	e msg	Embedded message	Varied	a	H+h
4	e xxxx	Embedded (inverted) 32-bit CRC (ASCII and Binary only). The embedded CRC is inverted so that the receiver does not recognize the embedded messages as messages to be output but continues with the RXCONFIG message. If you wish to use the messages output from the RXCONFIG log, simply flip the embedded CRC around for individual messages	Ulong	4	H+h+a
5	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+h+a+4
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.138 RXSTATUS
Receiver status

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log conveys various status parameters of the GNSS receiver system. These include the Receiver Status and Error words which contain several flags specifying status and error conditions. If an error occurs (shown in the Receiver Error word), the receiver idles all channels, turns off the antenna and disables the RF hardware as these conditions are considered to be fatal errors. The log contains a variable number of status words to allow for maximum flexibility and future expansion.

The receiver gives the user the ability to determine the importance of the status bits. In the case of the Receiver Status, setting a bit in the priority mask causes the condition to trigger an error. This causes the receiver to idle all channels, turn off the antenna and disable the RF hardware, the same as if a bit in the Receiver Error word is set. Setting a bit in an Auxiliary Status priority mask causes that condition to set the bit in the Receiver Status word corresponding to that Auxiliary Status. See also the STATUSCONFIG command on *page 319.*

ⓘ
1. Field #4, the receiver status word as represented in *Table 139*, is also in Field #8 of the header. See the *ASCII Example* below and *Table 139* on *page 661* for clarification.
2. Refer also to the chapter on *Built-In Status Tests* in the OEM6 Family Installation and Operation User Manual (OM-200000128).

Message ID: **93**
Log Type: **Asynch**

Recommended Input:

        log rxstatusa onchanged

Abbreviated ASCII Example:

```
#RXSTATUS COM1 0 90.5 FINESTEERING 1740 232531.278 00000020 2AE1 44913
00000000 4                               (RECEIVER ERROR)
00000020 00000000 00000000 00000000      (RECEIVER STATUS)
00040080 00001008 00000000 00000000      (AUX1 STATUS)
00000000 00000000 00000000 00000000      (AUX2 STATUS)
02000000 00000000 00000000 00000000      (AUX3 STATUS)
```

> ⓘ Receiver errors automatically generate event messages. These event messages
> are output in RXSTATUSEVENT logs. It is also possible to have status conditions
> trigger event messages to be generated by the receiver. This is done by setting/
> clearing the appropriate bits in the event set/clear masks. The set mask tells the
> receiver to generate an event message when the bit becomes set. Likewise, the
> clear mask causes messages to be generated when a bit is cleared. See the
> STATUSCONFIG command on *page 319* for details.
>
> If you wish to disable all these messages without changing the bits, simply
> UNLOG the RXSTATUSEVENT logs on the appropriate ports. See also the
> UNLOG command on *page 340*.

Table 138: Receiver Error

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Dynamic Random Access Memory (DRAM) status [a]	OK	Error
	1	0x00000002	Invalid firmware	OK	Error
	2	0x00000004	ROM status	OK	Error
	3	0x00000008	Reserved		
N1	4	0x00000010	Electronic Serial Number (ESN) access status	OK	Error
	5	0x00000020	Authorization code status	OK	Error
	6	0x00000040	Slow ADC could not be initialized	OK	Error
	7	0x00000080	Supply voltage status	OK	Error
N2	8	0x00000100	Thermometer could not be read	OK	Error
	9	0x00000200	Temperature status (as compared against acceptable limits)	OK	Error
	10	0x00000400	MINOS status	OK	Error
	11	0x00000800	PLL RF1 hardware status	OK	Error
N3	12	0x00001000	PLL RF2 hardware status	OK	Error
	13	0x00002000	PLL RF3 hardware status	OK	Error
	14	0x00004000	PLL RF4 hardware status	OK	Error
	15	0x00008000	NVM status	OK	Error
N4	16	0x00010000	Software resource limit	OK	Error
	17	0x00020000	Model not valid for this receiver	OK	Error
	18	0x00040000	Reserved		
	19	0x00080000			

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N5	20	0x00100000	Remote loading has begun	No	Yes
	21	0x00200000	Export restriction	OK	Error
	22	0x00400000	Safe Mode	OK	Error
	23	0x00800000	Reserved		
N6	24	0x01000000			
	25	0x02000000			
	26	0x04000000			
	27	0x08000000			
N7	28	0x10000000			
	29	0x20000000			
	30	0x40000000			
	31	0x80000000	Component hardware failure	OK	Error

a. RAM failure on an OEM6 card may also be indicated by a flashing red LED.

Table 139: Receiver Status

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Error flag, see *Table 138, Receiver Error* on *page 660*	No error	Error
	1	0x00000002	Temperature status	Within specifications	Warning
	2	0x00000004	Voltage supply status	OK	Warning
	3	0x00000008	Antenna power status See *ANTENNAPOWER* on *Page 70*	Powered	Not powered
N1	4	0x00000010	LNA Failure	OK	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 [b]	No overrun	Overrun
N3	12	0x00001000	Reserved		
	13	0x00002000	Aux transmit overrun flag	No overrun	Overrun
	14	0x00004000	AGC out of range[c]	OK	Out of range
	15	0x00008000	Reserved		

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N4	16	0x00010000	INS reset flag	No INS reset	INS 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, see *FIX* on *page 154*	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.

b. This flag indicates if any of the USB, ICOM, NCOM or XCOM ports are overrun. See the auxiliary status word for the specific port for which the buffer is overrun.

c. The AGC out of range (Bit 14) flag indicates if any of the RF AGCs are out of the range as indicated in *Table 140, Auxiliary 1 Status* on *page 663*.

Table 140: Auxiliary 1 Status

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Reserved		
	1	0x00000002			
	2	0x00000004			
	3	0x00000008	Position averaging	Off	On
N1	4	0x00000010	Reserved		
	5	0x00000020			
	6	0x00000040			
	7	0x00000080	USB connection status	Connected	Not connected
N2	8	0x00000100	USB1 buffer overrun flag	No overrun	Overrun
	9	0x00000200	USB2 buffer overrun flag	No overrun	Overrun
	10	0x00000400	USB3 buffer overrun flag	No overrun	Overrun
	11	0x00000800	Reserved		
N3	12	0x00001000	Profile Activation Bit = 0 (Ok) Bit = 1 (Error)		
	13	0x00002000	Reserved		
	14	0x00004000	RF1 AGC out of range	OK	Out of range
	15	0x00008000	RF2 AGC out of range	OK	Out of range
N4	16	0x00010000	RF3 AGC out of range	OK	Out of range
	17	0x00020000	RF4 AGC out of range	OK	Out of range
	18	0x00040000	Ethernet not connected	Connected	Not connected
	19	0x00080000	ICOM1 buffer overrun flag	No overrun	Overrun
N5	20	0x00100000	ICOM2 buffer overrun flag	No overrun	Overrun
	21	0x00200000	ICOM3 buffer overrun flag	No overrun	Overrun
	22	0x00400000	NCOM1 buffer overrun flag	No overrun	Overrun
	23	0x00800000	NCOM2 buffer overrun flag	No overrun	Overrun
N6	24	0x01000000	NCOM3 buffer overrun flag	No overrun	Overrun
	25	0x02000000	XCOM1 buffer overrun flag	No overrun	Overrun
	26	0x04000000	XCOM2 buffer overrun flag	No overrun	Overrun
	27	0x08000000	XCOM3 buffer overrun flag	No overrun	Overrun
N7	28	0x10000000	RF5 AGC out of range	OK	Out of range
	29	0x20000000	RF6 AGC out of range	OK	Out of range
	30	0x40000000	Reserved		
	31	0x80000000	Reserved		

Table 141: Auxiliary 2 Status

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	SPI Communication Failure	OK	Error
	1	0x00000002	I^2C Communication Failure	OK	Error
	2	0x00000004	COM4 buffer overrun flag	No overrun	Overrun
	3	0x00000008	COM5 buffer overrun flag	No overrun	Overrun
N1	4	0x00000010	COM6 buffer overrun flag	No overrun	Overrun
	5	0x00000020	COM7 buffer overrun flag	No overrun	Overrun
	6	0x00000040	COM8 buffer overrun flag	No overrun	Overrun
	7	0x00000080	COM9 buffer overrun flag	No overrun	Overrun
N2	8	0x00000100	COM10 buffer overrun flag	No overrun	Overrun
	9	0x00000200	Reserved		
	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	Reserved		
	25	0x02000000			
	26	0x04000000			
	27	0x08000000			
N7	28	0x10000000	Reserved		
	29	0x20000000			
	30	0x40000000			
	31	0x80000000			

Table 142: Auxiliary 3 Status

Nibble #	Bit #	Mask	Description[a]	Bit = 0	Bit = 1
N0	0	0x00000001	Reserved		
	1	0x00000002	WCOM1 buffer overrun flag	No overrun	Overrun
	2	0x00000004	FILE buffer overrun flag	No overrun	Overrun
	3	0x00000008	Bluetooth buffer overrun flag	No overrun	Overrun
N1	4	0x00000010	Reserved		
	5	0x00000020			
	6	0x00000040			
	7	0x00000080			
N2	8	0x00000100	Wi-Fi Power	Not powered	Powered
	9	0x00000200	Wi-Fi Client/AP Mode	Client mode	AP mode
	10	0x00000400	Wi-Fi Connection	connection not ready/ established - as client, not connected to AP - as AP, not ready for incoming connection	connection ready/ established - as client, connected to AP - as AP, ready for incoming connections
	11	0x00000800	Bluetooth Power	Not powered	Powered
N3	12	0x00001000	Bluetooth Pairing	Bluetooth SPP not connected	Bluetooth SPP connected
	13	0x00002000	Cell Radio Power	Not powered	Powered
	14	0x00004000	SIM Missing (unreadable)	Present and readable	Missing or unreadable
	15	0x00008000	SIM Locked	Unlocked	Locked
N4	16	0x00010000	Cell Roaming	Local	Roaming
	17	0x00020000	Cell Signal Service	Service available	No service available
	18	0x00040000	Reserved		
	19	0x00080000	Reserved		
N5	20	0x00100000	Reserved		
	21	0x00200000	Dual Antenna failed to power up	Dual antenna power OK	Dual antenna over current
	22	0x00400000	Internal Application Restarted	Application running	Application restarted
	23	0x00800000	Reserved		

Nibble #	Bit #	Mask	Description[a]	Bit = 0	Bit = 1
N6	24	0x01000000	Dual Card Connectivity	Device OK (if connected)	Communication lost
	25	0x02000000	VDC Out	VDC Out OK	VDC over current
	26	0x04000000	USB MST Error	USB stick OK (if inserted)	USB stick error
	27	0x08000000	Internal Storage Error	On board memory OK	On board memory error
N7	28	0x10000000	Reserved		
	29	0x20000000			
	30	0x40000000			
	31	0x80000000			

a. Some bits are only active when the corresponding modules are active. For example, the *BT Pairing* bit will not show *1 = device not paired* unless the Bluetooth module is enabled.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RXSTATUS header	Log header		H	0
2	error	Receiver error (see *Table 138, Receiver Error* on *page 660*). A value of zero indicates no errors	Ulong	4	H
3	# stats	Number of status codes (including Receiver Status). Each status code consists of 4 fields, the status, priority mask, event set mask and event clear mask. Each set is repeated for each status type. Note that for clarity, the Receiver Status, Auxiliary1 Status, Auxiliary 2 Status and Auxiliary3 Status is listed separately in this message	Ulong	4	H+4
4	rxstat	Receiver status word (see *Table 139, Receiver Status* on *page 661*)	Ulong	4	H+8
5	rxstat pri	Receiver status priority mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+12
6	rxstat set	Receiver status event set mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+16
7	rxstat clear	Receiver status event clear mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+20
8	aux1stat	Auxiliary 1 status word (see *Table 140, Auxiliary 1 Status* on *page 663*)	Ulong	4	H+24
9	aux1stat pri	Auxiliary 1 status priority mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+28

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
10	aux1stat set	Auxiliary 1 status event set mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+32
11	aux1stat clear	Auxiliary 1 status event clear mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+36
12	aux2stat	Auxiliary 2 status word (see *Table 141, Auxiliary 2 Status* on *page 664*)	Ulong	4	H+40
13	aux2stat pri	Auxiliary 2 status priority mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+44
14	aux2stat set	Auxiliary 2 status event set mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+48
15	aux2stat clear	Auxiliary 2 status event clear mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+52
16	aux3stat	Auxiliary 3 status word (see *Table 142, Auxiliary 3 Status* on *page 665*)	Ulong	4	H+56
17	aux3stat pri	Auxiliary 3 status priority mask, which can be set using the STATUSCONFIG command (see *page 319*)	Ulong	4	H+60
18	aux3stat set	Auxiliary 3 status event set mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+64
19	aux3stat clear	Auxiliary 3 status event clear mask, which can be set using the STATUSCONFIG command (*page 319*)	Ulong	4	H+68
20	xxxx	32-bit CRC (ASCII and Binary only)	ulong	4	H+72
21	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.139 RXSTATUSEVENT
Status event indicator

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log is used to output event messages as indicated in the RXSTATUS log (*page 659*). An event message is automatically generated for all receiver errors, which are indicated in the receiver error word. In addition, event messages can be generated when other conditions, which are indicated in the receiver status and auxiliary status words, are met. Whether or not an event message is generated under these conditions is specified using the STATUSCONFIG command (*page 319).*

On start-up, the receiver is set to log the RXSTATUSEVENTA log ONNEW on all ports. You can remove this message by using the UNLOG command, see *page 340.*

> See also the chapter on Built-In Status Tests in the OEM6 Family Installation and Operation User Manual (OM-20000128).

Message ID: **94**
Log Type: **Asynch**

Recommended Input:

        log rxstatuseventa onchanged

ASCII Example 1:

```
#RXSTATUSEVENTA,COM1,0,17.0,FREEWHEELING,1337,408334.510,00480000,
B967,1984;STATUS,19,SET,"NO VALID POSITION CALCULATED"*6DE945AD
```

ASCII Example 2:

```
#RXSTATUSEVENTA,COM1,0,41.0,FINESTEERING,1337,408832.031,01000400,
B967,1984;STATUS,10,SET,"COM3 TRANSMIT BUFFER OVERRUN"*5B5682A9
```

> When a fatal event occurs (for example, in the event of a receiver hardware failure), a bit is set in the receiver error word, part of the RXSTATUS log (*page 659*) to indicate the cause of the problem. Bit 0 is set in the receiver status word to show that an error occurred, the error strobe is driven high and the LED flashes red and yellow showing an error code. An RXSTATUSEVENT log is generated on all ports to show the cause of the error. Receiver tracking is disabled at this point but command and log processing continues to allow you to diagnose the error. Even if the source of the error is corrected at this point, the receiver must be reset to resume normal operation.

Table 143: Status Word

Word (binary)	Word (ASCII)	Description
0	ERROR	Receiver Error word, see *Table 138* on *page 660*
1	STATUS	Receiver Status word, see *Table 139* on *page 661*
2	AUX1	Auxiliary 1 Status word, see *Table 140* on *page 663*
3	AUX2	Auxiliary 2 Status word see *Table 141* on *page 664*
4	AUX3	Auxiliary 3 Status word see *Table 142* on *page 665*

Table 144: Event Type

Event		Description
Binary	ASCII	
0	CLEAR	Bit was cleared
1	SET	Bit was set

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RXSTATUSEVENT header	Log header		H	0
2	word	The status word that generated the event message (see *Table 143,* above)	Enum	4	H
3	bit position	Location of the bit in the status word (see *Table 139, Receiver Status* on *page 661* or the *Auxiliary Status* tables starting on *page 663*)	Ulong	4	H+4
4	event	Event type (see *Table 144* above)	Enum	4	H+8
3	description	This is a text description of the event or error	Char[32]	32	H+12
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.140 SATVIS
Satellite visibility

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains GPS and GLONASS satellite visibility data with additional satellite information.

> ⓘ
> 1. The SATVIS log is meant to provide a brief overview. The satellite positions and velocities used in the computation of this log are based on Almanac orbital parameters, not the higher precision Ephemeris parameters.
> 2. In the SATVIS log output there may be double satellite number entries. These are GLONASS antipodal satellites in the same orbit plane separated by 180 degrees latitude. Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.
> 3. The SATVIS log is generated every 10 seconds. If the log is requested at a faster rate than ontime 10, it will only be output every 10 seconds.

Message ID: **48**
Log Type: **Synch**

Recommended Input:

> log satvis ontime 60

Abbreviated ASCII Example:

```
<SATVIS COM1 0 81.5 FINESTEERING 1797 406620.000 00000000 0947 12942
<     TRUE TRUE 73
<          13 0 0 77.9 158.7 -763.045 -762.882
<          51 0 0 65.0 29.4 -1347.069 -1346.904
<          7 0 0 56.2 273.2 1887.773 1887.937
<          42 8 0 54.5 287.9 520.808 520.974
...
<          47 0 0 -73.5 213.1 919.905 920.071
<          12 0 0 -76.0 346.7 -784.609 -784.446
<          13 0 472 0.0 0.0 0.000 0.000
```

> ⓘ
> Consider sky visibility at each of the base and rover receivers in a differential setup. The accuracy and reliability of differential messages is proportional to the number of common satellites that are visible at the base and rover. Therefore, if the sky visibility at either station is poor, you might consider increasing the occupation times. This condition is best measured by monitoring the number of visible satellites during data collection along with the PDOP value (a value less than 3 is ideal). Also, the location and number of satellites in the sky is constantly changing. As a result, some periods in the day are slightly better for data collection than others. Use the SATVIS log to monitor satellite visibility. The PSRDOP log, see *page 578*, can be used to monitor the PDOP values.
>
> Site conditions surrounding the station that may affect satellite visibility and can generate noise in the data are water bodies, buildings, trees and nearby vehicles.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	SATVIS header	Log header		H	0
2	sat vis	Is satellite visibility valid? 0 = FALSE 1 = TRUE	Bool	4	H
3	comp alm	Was complete GPS almanac used? 0 = FALSE 1 = TRUE	Bool	4	H+4
4	#sat	Number of satellites with data to follow	Ulong	4	H+8
5	PRN/slot	Satellite PRN number of range measurement (GPS: 1-32 and SBAS: 120 to 138 or 183 to 187. For GLONASS, see *Section 1.3* on *page 34*)	Ushort	2	H+12
6	glofreq	(GLONASS Frequency + 7), see *Section 1.3* on *page 34*	Short	2	H+14
7	health	Satellite health [a]	Ulong	4	H+16
8	elev	Elevation (degrees)	Double	8	H+20
9	az	Azimuth (degrees)	Double	8	H+28
10	true dop	Theoretical Doppler of satellite - the expected Doppler frequency based on a satellite's motion relative to the receiver. It is computed using the satellite's coordinates and velocity along with the receiver's coordinates and velocity (Hz)	Double	8	H+36
11	app dop	Apparent Doppler for this receiver - the same as Theoretical Doppler above but with clock drift correction added (Hz)	Double	8	H+44
12	Next satellite offset = H + 12 + (#sat x 40)				
13	xxxx	32-bit CRC (ASCII and Binary only)	ulong	4	H+12+ (#sat x 40)
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. Satellite health values may be found in ICD-GPS-200. To obtain copies of ICD-GPS-200, refer to the Standards and References section of our website www.novatel.com/support/.

3.2.141 SATVIS2
Satellite visibility

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains satellite visibility data for all available systems with additional satellite and satellite system information. One log is output for each available satellite system.

> (i)
> 1. The SATVIS2 log is meant to provide a brief overview. The satellite positions and velocities used in the computation of this log are based on Almanac orbital parameters, not the higher precision Ephemeris parameters.
>
> 2. In the SATVIS2 log output there may be double satellite number entries. These are GLONASS antipodal satellites in the same orbit plane separated by 180 degrees latitude. Refer to the GLONASS section of our book An Introduction to GNSS, available on our website at www.novatel.com/support/.
>
> 3. The SATVIS2 log is generated every 10 seconds. If the log is requested at a faster rate than ontime 10, it will only be output every 10 seconds.

Message ID: **1043**
Log Type: **Asynch**

Recommended Input:

    log satvis2a onchanged

ASCII Example:

```
<SATVIS2 COM1 5 70.0 FINESTEERING 1729 166550.000 00000000 A867 44263
<     GPS TRUE TRUE 31
<         32 0 71.1 177.8 -1183.650 -1184.441
<         20 0 66.2 265.9 462.684 461.894
...
<         26 0 -78.7 246.3 805.272 804.481
<         9 0 -79.0 7.3 -930.480 -931.271
<SATVIS2 COM1 4 70.0 FINESTEERING 1729 166550.000 00000000 A867 44263
<     GLONASS TRUE TRUE 24
<         3+5 0 75.2 326.1 1088.078 1087.272
<         13-2 0 61.4 188.2 2243.727 2242.923
...
<         9-2 0 -72.3 6.3 -1384.534 -1385.337
<         7+5 0 -81.2 146.3 -666.742 -667.548
<SATVIS2 COM1 0 70.0 FINESTEERING 1729 166550.000 00000000 A867 44263
<     BEIDOU TRUE TRUE 14
<         11 0 2.6 342.2 -711.023 -711.807
<         12 0 -5.0 297.0 -2407.877 -2408.661
...
<         10 216 -79.3 254.5 122.316 121.532
<         13 216 -81.5 51.2 76.611 75.827
```

ⓘ Consider sky visibility at each of the base and rover receivers in a differential setup. The accuracy and reliability of differential messages is proportional to the number of common satellites that are visible at the base and rover. Therefore, if the sky visibility at either station is poor, you might consider increasing the occupation times. This condition is best measured by monitoring the number of visible satellites during data collection along with the PDOP value (a value less than 3 is ideal). Also, the location and number of satellites in the sky is constantly changing. As a result, some periods in the day are slightly better for data collection than others. Use the SATVIS2 log to monitor satellite visibility. The PSRDOP log can be used to monitor the PDOP values.

Site conditions surrounding the station that may affect satellite visibility and can generate noise in the data are water bodies, buildings, trees and nearby vehicles.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	SATVIS2 header	Log header		H	0
2	Satellite System	GNSS satellite system identifier. See *Table 110* on *page 508*	Enum	4	H
3	sat vis	Is satellite visibility valid? 0 = FALSE 1 = TRUE	Enum	4	H+4
4	comp alm	Was complete GNSS almanac used? 0 = FALSE 1 = TRUE	Enum	4	H+8
5	#sat	Number of satellites with data to follow	Ulong	4	H+12
6	Satellite ID	In binary logs, the satellite ID field is 4 bytes. The 2 lowest order bytes, interpreted as a USHORT, are the system identifier: for instance, the PRN for GPS or the slot for GLONASS. The 2 highest-order bytes are the frequency channel for GLONASS, interpreted as a SHORT and zero for all other systems. In ASCII and abbreviated ASCII logs, the satellite ID field is the system identifier. If the system is GLONASS and the frequency channel is not zero, then the signed channel is appended to the system identifier. For example, slot 13, frequency channel -2 is output as 13-2	Ulong	4	H+16
7	health	Satellite health [a]	Ulong	4	H+20
8	elev	Elevation (degrees)	Double	8	H+24
9	az	Azimuth (degrees)	Double	8	H+32

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
10	true dop	Theoretical Doppler of satellite - the expected Doppler frequency based on a satellite's motion relative to the receiver. It is computed using the satellite's coordinates and velocity along with the receiver's coordinates and velocity (Hz)	Double	8	H+40
11	app dop	Apparent Doppler for this receiver - the same as Theoretical Doppler above but with clock drift correction added (Hz)	Double	8	H+48
12	Next satellite offset = H + 16 + (#sat x 40)				
13	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+16+ (#sat x 40)
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. Satellite health values may be found in the applicable SIS-ICD for each system.

3.2.142 SATXYZ2
Satellite positions in ECEF Cartesian coordinates

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

When combined with a RANGE log, this data set contains the decoded satellite information necessary to compute the solution: satellite coordinates (ECEF WGS84), satellite clock correction, ionospheric corrections and tropospheric corrections. See the calculation examples in the usage box below. Only those satellites used in the corresponding PSRPOS solution are reported here. See also Figure 11, The WGS84 ECEF Coordinate System on *page 422*.

Message ID: **1451**
Log Type: **Synch**

Recommended Input:

         log satxyz2 ontime 1

Abbreviated ASCII Example:

```
<SATXYZ2 COM1 0 83.5 FINESTEERING 1686 489605.000 00000040 7513 43391
<     18
<     GPS 1 -15502299.3828 1012325.6443 21538404.8435 76246.262 6.990
      3.395 0.0 0.0
<     GPS 19 -25806091.5135 -6923139.1454 1709844.1975 -78547.421
      5.734 9.238 0.0 0.0
<     GPS 12 20368857.0090 -5772890.2153 15912912.0724 20118.104 2.415
      12.239 0.0 0.0
...
<     GLONASS 23+3 -22246787.0962 -4287240.2873 11721201.0046 -
      116210.453 6.928 4.205 0.0 0.0
<     GLONASS 7+5 4586441.8834 -14896106.2729 20222034.1193 -6061.174
      1.636 2.529 0.0 0.0
<     GLONASS 8+6 -12121452.4145 -4467306.1322 21995556.9720 -
      7165.609 0.350 2.586 0.0 0.0
```

The OEM6 family use positive numbers for ionospheric and tropospheric corrections. A positive clock offset indicates the clock is running ahead of the reference time. Positive ionospheric and tropospheric corrections are added to the geometric ranges or subtracted from the measured pseudoranges. For example:

P = p + pd + c(dT - dt) + d(ion) + d(trop) + Ep

is equivalent to

P - c(dT - dt) - d(ion) - d(trop) = p + pd + Ep

where

P = measured pseudorange
p = geometric range
pd = orbit error
dt = satellite clock offset
dT = receiver clock offset
d (ion) = ionospheric delay
d (trop) = tropospheric delay
c = speed of light
Ep = noise and multipath

Field #	Field Type	Description	Format	Binary Bytes	Binary Offset
1	SATXYZ2 header	Log header		H	0
2	#sat	Number of satellites to follow	Ulong	4	H
3	System	Satellite system (refer to *Table 110, Satellite System* on *page 508*)	Enum	4	H+4
4	Satellite ID	Satellite ID	Ulong	4	H+8
5	X	Satellite X co-ordinates (ECEF,m)	Double	8	H+12
6	Y	Satellite Y co-ordinates (ECEF,m)	Double	8	H+20
7	Z	Satellite Z co-ordinates (ECEF,m)	Double	8	H+28
8	clk corr	Satellite clock correction (m)	Double	8	H+36
9	iono delay	Ionosphere delay (m)	Double	8	H+44
10	tropo delay	Troposphere delay (m)	Double	8	H+52
11	Reserved1		Double	8	H+60
12	Reserved2		Double	8	H+68
13	Next satellite offset = H+4+(#sat x 72)				
14	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+(#sat x 72)
15	[CR][LF]	Sentence terminator (ASCII only)			

3.2.143 SBAS0
Do not use for safety applications

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This message tells you, that the GEO transmitting SBAS corrections is operating in test mode, and should not be used for safety applications for a period of time outlined in the SBAS signal specification.

> ▼ The GEO transmitting SBAS corrections is operating in test mode, and should not be used for safety-of-life applications.

See how the SBAS0 message relates to the SBAS testing modes in the SBASCONTROL command on *page 273*.

Message ID: **976**
Log Type: **Asynch**

Recommended Input:

      log sbas0a onchanged

ASCII Example:

```
#SBAS0A,COM1,0,68.5,SATTIME,1093,161299.000,00040020,7D6A,209;
122*E9A5AB08
```

> ⓘ Although the SBAS was designed for aviation users, it supports a wide variety of non-aviation uses including agriculture, surveying, recreation, and surface transportation.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	SBAS0 header	Log header		H	0
2	prn	Source PRN message - also PRN not to use	Ulong	4	H
3	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4
4	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.144 SBAS1
PRN mask assignments

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The PRN mask is given in SBAS1. The transition of the PRN mask to a new one (which will be infrequent) is controlled with the 2-bit IODP, which sequences to a number between 0 and 3. The same IODP appears in the applicable SBAS2, SBAS3, SBAS4, SBAS5, SBAS7, SBAS24 and SBAS25 messages (SBAS32, SBAS33, SBAS34, SBAS35 and SBAS45). This transition would probably only occur when a new satellite is launched or when a satellite fails and is taken out of service permanently. A degraded satellite may be flagged as a "don't use" satellite temporarily.

Message ID: **977**
Log Type: **Asynch**

Recommended Input:

> log sbas1a onchanged

ASCII Example:

```
#SBAS1A,COM1,0,24.5,SATTIME,1337,415802.000,00000000,5955,1984;
134,FFEFFFFE000000000000000000000004004000000000000000000000,
2*3633CF7B
```

> ⓘ   Each raw SBAS frame gives data for a specific frame decoder number. The SBAS1 message can be logged to view the data breakdown of SBAS frame 1 which contains information about the PRN mask assignment.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	header	Log header		H	0
2	prn	Source PRN of message	Ulong	4	H
3	mask	PRN bit mask	Uchar[27]	28 [a]	H+4
4	iodp	Issue of PRN mask data	Ulong	4	H+32
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+36
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary log case, an additional 1 byte of padding is added to maintain 4-byte alignment.

3.2.145 SBAS2
Fast correction slots 0-12

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS2 are fast corrections for slots 0-12 in the mask of SBAS1. This message may or may not come when SBAS is in testing mode (see the `SBASCONTROL` command on *page 273* for details).

Message ID: **982**
Log Type: **Asynch**

Recommended Input:

> log sbas2a onchanged

ASCII Example:

```
#SBAS2A,COM1,0,29.0,SATTIME,1337,415925.000,00000000,E194,1984;
134,2,2,3,-3,5,1,2047,-2,2047,2047,2047,2047,2047,-3,2,5,11,7,8,
14,8,14,14,14,14,14,6,12*8D8D2E1C
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS2 message can be logged to view the data breakdown of SBAS frame 2 which contains information about fast correction slots 0-12.

Table 145: Evaluation of UDREI

UDREI [a]	UDRE metres	σ^2 i.udre metres^2
0	0.75	0.0520
1	1.0	0.0924
2	1.25	0.1444
3	1.75	0.2830
4	2.25	0.4678
5	3.0	0.8315
6	3.75	1.2992
7	4.5	1.8709
8	5.25	2.5465
9	6.0	3.3260
10	7.5	5.1968
11	15.0	20.7870
12	50.0	230.9661
13	150.0	2078.695
14	Not Monitored	Not Monitored
15	Do Not Use	Do Not Use

a. The σ^2UDRE broadcast in SBAS2, SBAS3, SBAS4, SBAS5, SBAS6 and SBAS24 applies at a time prior to or at the time of applicability of the associated corrections.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS2 header	Log header		H	0	
2	prn	Source PRN of message	Ulong	4	H	-
3	iodf	Issue of fast corrections data	Ulong	4	H+4	-
4	iodp	Issue of PRN mask data	Ulong	4	H+8	-
5	prc0		Long	4	H+12	-
6	prc1		Long	4	H+16	-
7	prc2		Long	4	H+20	-
8	prc3		Long	4	H+24	-
9	prc4		Long	4	H+28	-
10	prc5	prc(i):	Long	4	H+32	-
11	prc6	Fast corrections	Long	4	H+36	-
12	prc7	(-2048 to +2047) for the prn in slot i (i = 0-12)	Long	4	H+40	-
13	prc8		Long	4	H+44	-
14	prc9		Long	4	H+48	-
15	prc10		Long	4	H+52	-
16	prc11		Long	4	H+56	-
17	prc12		Long	4	H+60	-
18	udre0		Ulong	4	H+64	
19	udre1		Ulong	4	H+68	
20	udre2		Ulong	4	H+72	
21	udre3		Ulong	4	H+76	
22	udre4		Ulong	4	H+80	
23	udre5	udre(i):	Ulong	4	H+84	See *Table 145, Evaluation of UDREI* on *page 679*
24	udre6	User differential range error indicator for the prn in slot i (i = 0-12)	Ulong	4	H+88	
25	udre7		Ulong	4	H+92	
26	udre8		Ulong	4	H+96	
27	udre9		Ulong	4	H+100	
28	udre10		Ulong	4	H+104	
29	udre11		Ulong	4	H+108	
30	udre12		Ulong	4	H+112	
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+116	-
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.146 SBAS3
Fast corrections slots 13-25

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS3 are fast corrections for slots 13-25 in the mask of SBAS1.

Message ID: **987**
Log Type: **Asynch**

Recommended Input:

> log sbas3a onchanged

ASCII Example:

```
#SBAS3A,COM1,0,17.0,SATTIME,1337,415990.000,00000000,BFF5,1984;
134,1,2,2047,0,2047,2047,-21,-4,2047,2047,-1,0,2,2047,6,14,5,14,
14,11,5,14,14,5,7,5,14,8*A25AEBC5
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS3 message can be logged to view the data breakdown of SBAS frame 3 which contains information about fast correction slots 13-25.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS3 header	Log header		H	0	
2	prn	Source PRN of message	Ulong	4	H	-
3	iodf	Issue of fast corrections data	Ulong	4	H+4	-
4	iodp	Issue of PRN mask data	Ulong	4	H+8	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
5	prc13		Long	4	H+12	-
6	prc14		Long	4	H+16	-
7	prc15		Long	4	H+20	-
8	prc16		Long	4	H+24	-
9	prc17		Long	4	H+28	-
10	prc18	prc(i):	Long	4	H+32	-
11	prc19		Long	4	H+36	-
12	prc20	Fast corrections (-2048 to +2047) for the prn in slot i (i = 13-25)	Long	4	H+40	-
13	prc21		Long	4	H+44	-
14	prc22		Long	4	H+48	-
15	prc23		Long	4	H+52	-
16	prc24		Long	4	H+56	-
17	prc25		Long	4	H+60	-
18	udre13		Ulong	4	H+64	
19	udre14		Ulong	4	H+68	
20	udre15		Ulong	4	H+72	
21	udre16		Ulong	4	H+76	
22	udre17		Ulong	4	H+80	
23	udre18	udre(i):	Ulong	4	H+84	See *Table 145, Evaluation of UDREI* on *page 679*
24	udre19	User differential range error indicator for the prn in slot i (i = 13-25)	Ulong	4	H+88	
25	udre20		Ulong	4	H+92	
26	udre21		Ulong	4	H+96	
27	udre22		Ulong	4	H+100	
28	udre23		Ulong	4	H+104	
29	udre24		Ulong	4	H+108	
30	udre25		Ulong	4	H+112	
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+116	-
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.147 SBAS4
Fast correction slots 26-38

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS4 are fast corrections for slots 26-38 in the mask of SBAS1.

Message ID: **992**
Log Type: **Asynch**

Recommended Input:

    log sbas4a onchanged

ASCII Example:

```
#SBAS4A,COM1,0,58.0,SATTIME,1093,163399.000,00000020,B4B0,209;
122,0,3,2047,3,-1,2047,2047,2047,-3,-1,5,3,3,2047,2,14,3,3,14,
14,14,6,3,4,5,4,14,3*2E0894B1
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The
>   SBAS4 message can be logged to view the data breakdown of SBAS frame 4
>   which contains information about fast correction slots 26-38.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS4 header	Log header		H	0	
2	prn	Source PRN of message	Ulong	4	H	-
3	iodf	Issue of fast corrections data	Ulong	4	H+4	-
4	iodp	Issue of PRN mask data	Ulong	4	H+8	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
5	prc26		Long	4	H+12	-
6	prc27		Long	4	H+16	-
7	prc28		Long	4	H+20	-
8	prc29		Long	4	H+24	-
9	prc30		Long	4	H+28	-
10	prc31	prc(i):	Long	4	H+32	-
11	prc32		Long	4	H+36	-
12	prc33	Fast corrections (-2048 to +2047) for the prn in slot i (i = 26-38)	Long	4	H+40	-
13	prc34		Long	4	H+44	-
14	prc35		Long	4	H+48	-
15	prc36		Long	4	H+52	-
16	prc37		Long	4	H+56	-
17	prc38		Long	4	H+60	-
18	udre26		Ulong	4	H+64	
19	udre27		Ulong	4	H+68	
20	udre28		Ulong	4	H+72	
21	udre29		Ulong	4	H+76	
22	udre30		Ulong	4	H+80	
23	udre31	udre(i):	Ulong	4	H+84	See *Table 145, Evaluation of UDREI* on *page 679*
24	udre32	User differential range error indicator for the prn in slot i (i = 26-38)	Ulong	4	H+88	
25	udre33		Ulong	4	H+92	
26	udre34		Ulong	4	H+96	
27	udre35		Ulong	4	H+100	
28	udre36		Ulong	4	H+104	
29	udre37		Ulong	4	H+108	
30	udre38		Ulong	4	H+112	
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+116	-
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.148 SBAS5
Fast correction slots 39-50

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS5 are fast corrections for slots 39-50 in the mask of SBAS1.

Message ID: **994**
Log Type: **Asynch**

Recommended Input:

> log sbas5a onchanged

ASCII Example:

```
#SBAS5A,COM1,0,72.5,SATTIME,1093,161480.000,00040020,31D4,209;122,
1,3,-7,2047,2047,2047,-4,2047,2047,2047,9,2047,2047,-3,-2,11,14,14,
14,4,14,14,14,5,14,14,4,2*2BF0109B
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS5 message can be logged to view the data breakdown of SBAS frame 5 which contains information about fast correction slots 39-50.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS5 header	Log header		H	0	
2	prn	Source PRN of message	Ulong	4	H	-
3	iodf	Issue of fast corrections data	Ulong	4	H+4	-
4	iodp	Issue of PRN mask data	Ulong	4	H+8	-
5	prc39		Long	4	H+12	-
6	prc40		Long	4	H+16	-
7	prc41		Long	4	H+20	-
8	prc42		Long	4	H+24	-
9	prc43		Long	4	H+28	-
10	prc44	prc(i):	Long	4	H+32	-
11	prc45	Fast corrections (-2048 to +2047) for the prn in slot i (i = 39-50)	Long	4	H+36	-
12	prc46		Long	4	H+40	-
13	prc47		Long	4	H+44	-
14	prc48		Long	4	H+48	-
15	prc49		Long	4	H+52	-
16	prc50		Long	4	H+56	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
17	prc51 (Invalid, do not use)		Long	4	H+60	-
18	udre39		Ulong	4	H+64	
19	udre40		Ulong	4	H+68	
20	udre41		Ulong	4	H+72	
21	udre42		Ulong	4	H+76	
22	udre43	udre(i):	Ulong	4	H+80	
23	udre44		Ulong	4	H+84	See *Table 145, Evaluation of UDREI* on page 679
24	udre45	User differential range error indicator for the prn in slot i (i = 39-50)	Ulong	4	H+88	
25	udre46		Ulong	4	H+92	
26	udre47		Ulong	4	H+96	
27	udre48		Ulong	4	H+100	
28	udre49		Ulong	4	H+104	
29	udre50		Ulong	4	H+108	
30	udre51 (Invalid, do not use)		Ulong	4	H+112	
31	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+116	-
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.149 SBAS6

Integrity message

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS6 is the integrity information message. Each message includes an IODF for each fast corrections message. The σ^2_{UDRE} information for each block of satellites applies to the fast corrections with the corresponding IODF.

Message ID: **995**
Log Type: **Asynch**

Recommended Input:

> log sbas6a onchanged

ASCII Example:

```
#SBAS6A,COM1,0,57.5,SATTIME,1093,273317.000,00000020,526A,209;
122,3,3,3,3,9,14,14,2,3,10,2,14,14,3,14,14,5,14,14,7,14,14,14,14,
14,14,3,3,14,14,14,14,3,15,11,11,15,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0*925A2A9B
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS6 message can be logged to view the data breakdown of SBAS frame 6 which contains information about the integrity message.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS6 header	Log header		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	iodf2	Issue of fast corrections data	Ulong	4	H+4	-
4	iodf3	Issue of fast corrections data	Ulong	4	H+8	-
5	iodf4	Issue of fast corrections data	Ulong	4	H+12	-
6	iodf5	Issue of fast corrections data	Ulong	4	H+16	-
7	udre0	udre(i): User differential range error indicator for the prn in slot i (i = 0-50)	Ulong	4	H+20	See *Table 145, Evaluation of UDREI* on *page 679*
8	udre1		Ulong	4	H+24	
9	udre2		Ulong	4	H+28	
10	udre3		Ulong	4	H+32	
11	udre4		Ulong	4	H+36	
12	udre5		Ulong	4	H+40	

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
13	udre6		Ulong	4	H+44	
14	udre7		Ulong	4	H+48	
15	udre8		Ulong	4	H+52	
16	udre9		Ulong	4	H+56	
17	udre10		Ulong	4	H+60	
18	udre11		Ulong	4	H+64	
19	udre12		Ulong	4	H+68	
20	udre13		Ulong	4	H+72	
21	udre14		Ulong	4	H+76	
22	udre15		Ulong	4	H+80	
23	udre16		Ulong	4	H+84	
24	udre17		Ulong	4	H+88	
25	udre18		Ulong	4	H+92	See *Table 145, Evaluation of UDREI* on *page 679*
26	udre19	udre(i):	Ulong	4	H+96	
27	udre20		Ulong	4	H+100	
28	udre21	User differential range error indicator for the prn in slot i	Ulong	4	H+104	
29	udre22	(i = 0-50)	Ulong	4	H+108	
30	udre23		Ulong	4	H+112	
31	udre24		Ulong	4	H+116	
32	udre25		Ulong	4	H+120	
33	udre26		Ulong	4	H+124	
34	udre27		Ulong	4	H+128	
35	udre28		Ulong	4	H+132	
36	udre29		Ulong	4	H+136	
37	udre30		Ulong	4	H+140	
38	udre31		Ulong	4	H+144	
39	udre32		Ulong	4	H+148	
40	udre33		Ulong	4	H+152	
41	udre34		Ulong	4	H+156	
42	udre35		Ulong	4	H+160	
43	udre36		Ulong	4	H+164	
44	udre37		Ulong	4	H+168	

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
45	udre38		Ulong	4	H+172	
46	udre39		Ulong	4	H+176	
47	udre40		Ulong	4	H+180	
48	udre41		Ulong	4	H+184	
49	udre42		Ulong	4	H+188	
50	udre43		Ulong	4	H+192	
51	udre44		Ulong	4	H+196	
52	udre45		Ulong	4	H+200	
53	udre46	udre(i): User differential range error indicator for the prn in slot i (i = 0-50)	Ulong	4	H+204	See *Table 145, Evaluation of UDREI* on *page 679*
54	udre47		Ulong	4	H+208	
55	udre48		Ulong	4	H+212	
56	udre49		Ulong	4	H+216	
58	udre50		Ulong	4	H+220	
58	udre51 (Invalid, do not use)		Ulong	4	H+224	
59	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+228	-
60	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.150 SBAS7

Fast correction degradation

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The SBAS7 message specifies the applicable IODP, system latency time and fast degradation factor indicator for computing the degradation of fast and long term corrections.

Message ID: **996**
Log Type: **Asynch**

Recommended Input:

       log sbas7a onchanged

ASCII Example:

```
#SBAS7A,COM1,0,36.5,SATTIME,1337,416367.000,00000000,12E3,1984;
122,1,2,0,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,
15,15,15,15,15,15,15,15,15,15,15,15,15,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0*827A7364
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS7 message can be logged to view the data breakdown of SBAS frame 7 which contains information about fast correction degradation.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	SBAS7 header	Log header		H	0
2	prn	Source PRN of message	Ulong	4	H
3	latency	System latency (s)	Ulong	4	H+4
4	iodp	Issue of PRN mask data	Ulong	4	H+8
5	spare bits	Unused spare bits	Ulong	4	H+12
6	aI(0)	aI(i): Degradation factor indicator for the prn in slot i (i = 0-50)	Ulong	4	H+16
7	aI(1)		Ulong	4	H+20
8	aI(2)		Ulong	4	H+24
9	aI(3)		Ulong	4	H+28
10	aI(4)		Ulong	4	H+32
11	aI(5)		Ulong	4	H+36
12	aI(6)		Ulong	4	H+40
13	aI(7)		Ulong	4	H+44
14	aI(8)		Ulong	4	H+48

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
15	al(9)		Ulong	4	H+52
16	al(10)		Ulong	4	H+56
17	al(11)		Ulong	4	H+60
18	al(12)		Ulong	4	H+64
19	al(13)		Ulong	4	H+68
20	al(14)		Ulong	4	H+72
21	al(15)		Ulong	4	H+76
22	al(16)		Ulong	4	H+80
23	al(17)		Ulong	4	H+84
24	al(18)		Ulong	4	H+88
25	al(19)		Ulong	4	H+92
26	al(20)		Ulong	4	H+96
27	al(21)		Ulong	4	H+100
28	al(22)	al(i):	Ulong	4	H+104
29	al(23)	Degradation factor indicator for the prn in slot i (i = 0-50)	Ulong	4	H+108
30	al(24)		Ulong	4	H+112
31	al(25)		Ulong	4	H+116
32	al(26)		Ulong	4	H+120
33	al(27)		Ulong	4	H+124
34	al(28)		Ulong	4	H+128
35	al(29)		Ulong	4	H+132
36	al(30)		Ulong	4	H+136
37	al(31)		Ulong	4	H+140
38	al(32)		Ulong	4	H+144
39	al(33)		Ulong	4	H+148
40	al(34)		Ulong	4	H+152
41	al(35)		Ulong	4	H+156
42	al(36)		Ulong	4	H+160
43	al(37)		Ulong	4	H+164
44	al(38)		Ulong	4	H+168
45	al(39)		Ulong	4	H+172
46	al(40)		Ulong	4	H+176

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
47	al(41)		Ulong	4	H+180
48	al(42)		Ulong	4	H+184
49	al(43)		Ulong	4	H+188
50	al(44)		Ulong	4	H+192
51	al(45)		Ulong	4	H+196
52	al(46)		Ulong	4	H+200
53	al(47)		Ulong	4	H+204
54	al(48)		Ulong	4	H+208
55	al(49)	al(i):	Ulong	4	H+212
56	al(50)	Degradation factor indicator for the prn in slot i (i = 0-50)	Ulong	4	H+216
57	al(51) (Invalid, do not use)		Ulong	4	H+220
58	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+224
59	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.151 SBAS9
GEO navigation message

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS9 provides the GEO navigation message representing the position, velocity and acceleration of the geostationary satellite, in ECEF coordinates and its apparent clock time and frequency offsets.

Also included is the time of applicability, an Issue of Data (IOD) and an accuracy exponent (URA) representing the estimated accuracy of the message. The time offset and time drift are with respect to SBAS Network Time. Their combined effect is added to the estimate of the satellite's transmit time.

Message ID: **997**
Log Type: **Asynch**

Recommended Input:

> log sbas9a onchanged

ASCII Example:

```
#SBAS9A,COM1,0,38.0,SATTIME,1337,416426.000,00000000,B580,1984;
122,175,70848,2,24802064.1600,-34087313.9200,-33823.2000,
1.591250000,0.107500000,0.6080000,-0.0000750,-0.0001125,
0.000187500,-2.235174179E-08,9.094947018E-12*636051D2
```

> ⓘ Each raw SBAS frame gives data for a specific frame decoder number. The SBAS9 message can be logged to view the data breakdown of SBAS frame 9 which contains the GEO navigation message.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	SBAS9 header	Log header		H	0
2	prn	Source PRN of message	Ulong	4	H
3	iodn	Issue of GEO navigation data	Ulong	4	H+4
4	t_0	Time of applicability	Ulong	4	H+8
5	ura	URA value	Ulong	4	H+12
6	x	ECEF x coordinate (m)	Double	8	H+16
7	y	ECEF y coordinate (m)	Double	8	H+24
8	z	ECEF z coordinate (m)	Double	8	H+32
9	xvel	X rate of change (m/s)	Double	8	H+40
10	yvel	Y rate of change (m/s)	Double	8	H+48
11	zvel	Z rate of change (m/s)	Double	8	H+56

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
12	xaccel	X rate of rate change (m/s2)	Double	8	H+64
13	yaccel	Y rate of rate change (m/s2)	Double	8	H+72
14	zaccel	Z rate of rate change (m/s2)	Double	8	H+80
15	a_{f0}	Time offset (s)	Double	8	H+88
16	a_{f1}	Time drift (s)	Double	8	H+96
17	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+104
18	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.152 SBAS10
Degradation factor

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The fast corrections, long term corrections and ionospheric corrections are all provided in the SBAS10 message.

Message ID:	**978**
Log Type:	**Asynch**

Recommended Input:

```
log SBAS10a onchanged
```

ASCII Example:

```
#SBAS10A,COM1,0,35.5,SATTIME,1337,416469.000,00000000,C305,1984;
122,54,38,76,256,152,100,311,83,256,6,0,300,292,0,1,000000000000000
0000000*8884D248
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS10 message can be logged to view the data breakdown of SBAS frame 10 which contains information about degradation factors.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS10 header	Log header		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	b_{rcc}	Estimated noise and round off error parameter	Ulong	4	H+4	0.002
4	c_{ltc_lsb}	Maximum round off due to the least significant bit (lsb) of the orbital clock	Ulong	4	H+8	0.002
5	c_{ltc_vl}	Velocity error bound	Ulong	4	H+12	0.00005
6	i_{ltc_vl}	Update interval for v=1 long term	Ulong	4	H+16	-
7	c_{ltc_v0}	Bound on update delta	Ulong	4	H+20	0.002
8	i_{ltc_v1}	Minimum update interval v = 0	Ulong	4	H+24	-
9	c_{geo_lsb}	Maximum round off due to the lsb of the orbital clock	Ulong	4	H+28	0.0005
10	c_{geo_v}	Velocity error bound	Ulong	4	H+32	0.00005
11	i_{geo}	Update interval for GEO navigation message	Ulong	4	H+36	-
12	c_{er}	Degradation parameter	Ulong	4	H+40	0.5

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
13	c_{iono_step}	Bound on ionospheric grid delay difference	Ulong	4	H+44	0.001
14	i_{iono}	Minimum ionospheric update interval	Ulong	4	H+48	-
15	c_{iono_ramp}	Rate of ionospheric corrections change	Ulong	4	H+52	0.000005
16	rss_{udre}	User differential range error flag	Ulong	4	H+56	-
17	rss_{iono}	Root sum square flag	Ulong	4	H+60	-
18	spare bits	Spare 88 bits, possibly GLONASS	Hex[11]	11	H+64	-
19	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+75	-
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.153 SBAS12
SBAS network time and UTC

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS12 contains information bits for the UTC parameters and UTC time standard from which an offset is determined. The UTC parameters correlate UTC time with the SBAS network time rather than with GPS reference time.

Message ID: **979**
Log Type: **Asynch**

Recommended Input:

> log sbas12a onchanged

> ⓘ Each raw SBAS frame gives data for a specific frame decoder number. The SBAS12 message can be logged to view the data breakdown of SBAS frame 12 which contains information about time parameters.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	SBAS12 header	Log header		H	0
2	prn	Source PRN of message	Ulong	4	H
3	A_1	Time drift (s/s)	Double	8	H+4
4	A_0	Time offset (s)	Double	8	H+12
5	seconds	Seconds into the week (s)	Ulong	4	H+20
6	week	Week number	Ushort	4	H+24
7	dt_{ls}	Delta time due to leap seconds	Short	2	H+26
8	wn_{lsf}	Week number, leap second future	Ushort	2	H+28
9	dn	Day of the week (the range is 1 to 7 where Sunday = 1 and Saturday = 7)	Ushort	2	H+30
10	dt_{lsf}	Delta time, leap second future	Ushort	2	H+32
11	utc id	UTC type identifier	Ushort	2	H+34
12	gpstow	GPS reference time of the week	Ulong	4	H+36
13	gpswn	GPS de-modulo week number	Ulong	4	H+40
14	glo indicator	Is GLONASS information present? 0 = FALSE 1 = TRUE	Enum	4	H+44
15	Reserved array of hexabytes for GLONASS		Char[10]	12[a]	H+48
16	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+60
17	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

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

3.2.154 SBAS17
GEO Almanac message

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Almanacs for all GEOs are broadcast periodically to alert you of their existence, location, the general service provided, status and health.

> ⓘ Unused almanacs have a PRN number of 0 and should be ignored, see *ASCII Example* below.

Message ID: 980
Log Type: Asynch

Recommended Input:

       log sbas17a onchanged

ASCII Example:

```
#SBAS17A,COM1,0,84.5,SATTIME,1610,514149.000,00000000,896C,39061;
135,3,0,135,0,-11536200,-40536600,-260000,0,0,0,0,138,0,-12521600,
-40258400, 0,0,0,0,0,133,0,-5551000,-41774200,-1248000,0,0,120,
82112*2BE5146F
```

> ⓘ Each raw SBAS frame gives data for a specific frame decoder number. The SBAS17 message can be logged to view the data breakdown of SBAS frame 17 which contains GEO almanacs.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS17 header	Log header		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	#ents	Number of almanac entries with information to follow	Ulong	4	H+4	-
4	data id	Data ID type	Ushort	2	H+8	-
5	entry prn	PRN for this entry	Ushort	2	H+10	-
6	health	Health bits	Ushort	4[a]	H+12	-
7	x	ECEF x coordinate (m)	Long	4	H+16	-
8	y	ECEF y coordinate (m)	Long	4	H+20	-
9	z	ECEF z coordinate (m)	Long	4	H+24	-
10	x vel	X rate of change (m/s)	Long	4	H+28	-
11	y vel	Y rate of change (m/s)	Long	4	H+32	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
12	z vel	Z rate of change (m/s)	Long	4	H+36	-
13...	Next entry = H+8+(#ents x 32)					-
14	t0	Time of day in seconds (0 to 86336)	Ulong	4	H+8+ (#ents x 32)	64
15	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+12+ (#ents x 32)	-
16	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

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

3.2.155 SBAS18
IGP mask

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The ionospheric delay corrections are broadcast as vertical delay estimates, at specified ionospheric grid points (IGPs), applicable to a signal on L1. The predefined IGPs are contained in 11 bands (numbered 0 to 10). Bands 0-8 are vertical bands on a Mercator projection map and bands 9-10 are horizontal bands on a Mercator projection map. Since it is impossible to broadcast IGP delays for all possible locations, a mask is broadcast to define the IGP locations providing the most efficient model of the ionosphere at the time.

Message ID: **981**
Log Type: **Asynch**

Recommended Input:

> log sbas18a onchanged

ASCII Example:

```
#SBAS18A,COM1,0,33.0,SATTIME,1337,417074.000,00000000,F2C0,1984;
122,4,2,2,0000FFC0007FC0003FF0000FF80007FE0007FE0003FF0000FF80,
0*B1ED353E
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS18 message can be logged to view the data breakdown of SBAS frame 18 which contains information about ionospheric grid points.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	SBAS18 header	Log header		H	0
2	prn	Source PRN of message	Ulong	4	H
3	#bands	Number of bands broadcast	Ulong	4	H+4
4	band num	Specific band number that identifies which of the 11 IGP bands the data belongs to	Ulong	4	H+8
5	iodi	Issue of ionospheric data	Ulong	4	H+12
6	igp mask	IGP mask	Uchar[26]	28[a]	H+16
7	spare bit	One spare bit	Ulong	4	H+44
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+48
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

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

3.2.156 SBAS24
Mixed fast/slow corrections

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

If there are 6 or fewer satellites in a block, they may be placed in this mixed correction message. There is a fast data set for each satellite and a UDRE indicator. Each message also contains an IODP indicating the associated PRN mask.

The fast correction (PRC) has a valid range of -2048 to +2047. If the range is exceeded, a don't use indication is inserted into the user differential range error indicator (UDREI) field, see *Table 145* on *page 679*. You should ignore extra data sets not represented in the PRN mask.

The time of applicability (T0) of the PRC is the start of the epoch of the WNT second that is coincident with the transmission at the GEO satellite of the first bit of the message block.

Message ID: 983
Log Type: Asynch

Recommended Input:

        log sbas24a onchanged

ASCII Example:

```
#SBAS24A,COM1,0,34.0,SATTIME,1337,417108.000,00000000,0A33,1984;
134,2047,2047,2047,2047,-1,-2,14,14,14,14,11,14,2,2,0,0,1,0,0,0,
0,0,0,0,0,0,0,0,0,0,0*76FF954B
```

> ⓘ Each raw SBAS frame gives data for a specific frame decoder number. The SBAS24 message can be logged to view the data breakdown of SBAS frame 24 which contains mixed fast/slow corrections.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS24 header	Log header		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	prc0	prc(i): Fast corrections (-2048 to +2047) for the prn in slot i (i = 0-5)	Long	4	H+4	-
4	prc1		Long	4	H+8	-
5	prc2		Long	4	H+12	-
6	prc3		Long	4	H+16	-
7	prc4		Long	4	H+20	-
8	prc5		Long	4	H+24	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
9	udre0	udre(i): User differential range error indicator for the prn in slot i (i = 0-5)	Ulong	4	H+28	See *Table 145* on *page 679*
10	udre1		Ulong	4	H+32	
11	udre2		Ulong	4	H+36	
12	udre3		Ulong	4	H+40	
13	udre4		Ulong	4	H+44	
14	udre5		Ulong	4	H+48	
15	iodp	Issue of PRN mask data	Ulong	4	H+52	-
16	block id	Associated message type	Ulong	4	H+56	
17	iodf	Issue of fast corrections data	Ulong	4	H+60	-
18	spare	Spare value	Ulong	4	H+64	-
19	vel	Velocity code flag	Ulong	4	H+68	-
20	mask1	Index into PRN mask (Type 1)	Ulong	4	H+72	-
21	iode1	Issue of ephemeris data	Ulong	4	H+76	-
22	dx1	Delta x (ECEF)	Long	4	H+80	0.125
23	dy1	Delta y (ECEF)	Long	4	H+84	0.125
24	dz1	Delta z (ECEF)	Long	4	H+88	0.125
25	da^{f0}	Delta a^{f0} clock offset	Long	4	H+92	2^{-31}
26	mask2	Second index into PRN mask (Type 1)	Ulong	4	H+96	-
27	iode2	Second issue of ephemeris data	Ulong	4	H+100	-
28	ddx	Delta delta x (ECEF)	Long	4	H+104	2^{-11}
29	ddy	Delta delta y (ECEF)	Long	4	H+108	2^{-11}
30	ddz	Delta delta z (ECEF)	Long	4	H+112	2^{-11}
31	da^{f1}	Delta a^{f1} clock offset	Long	4	H+116	2^{-39}
32	t_0	Applicable time of day	Ulong	4	H+120	16
33	iodp	Issue of PRN mask data	Ulong	4	H+124	-
34	corr spare	Spare value when velocity code is equal to 0	Ulong	4	H+128	-
35	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+132	-
36	[CR][LF]	Sentence terminator (ASCII only)	-	-	H+136	-

3.2.157 SBAS25
Long term slow satellite corrections

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS25 provides error estimates for slow varying satellite ephemeris and clock errors with respect to WGS-84 ECEF coordinates.

Message ID: **984**
Log Type: **Asynch**

Recommended Input:

       log sbas25a onchanged

ASCII Example:
```
#SBAS25A,COM1,0,37.5,SATTIME,1337,417193.000,00000000,B8FF,1984;
134,1,19,25,-1,-3,0,-15,0,0,0,1,-1,-2,4465,2,0,1,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0*81685317
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS25 message can be logged to view the data breakdown of SBAS frame 25 which contains long term slow satellite corrections.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS25 header	Log header		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	1st half vel	Velocity code flag (0 or 1)	Ulong	4	H+4	-
4	1st half mask1	Index into PRN mask (Type 1)	Ulong	4	H+8	-
5	1st half iode1	Issue of ephemeris data	Ulong	4	H+12	-
6	1st half dx1	Delta x (ECEF)	Long	4	H+16	0.125
7	1st half dy1	Delta y (ECEF)	Long	4	H+20	0.125
8	1st half dz1	Delta z (ECEF)	Long	4	H+24	0.125
9	1st half a_{f0}	Delta a_{f0} clock offset	Long	4	H+28	2^{-31}
10	1st half mask2	Second index into PRN mask (Type 1) Dummy value when velocity code = 1	Ulong	4	H+32	-
11	1st half iode2	Second issue of ephemeris data Dummy value when velocity code = 1	Ulong	4	H+36	-
12	1st half ddx	Delta delta x (ECEF) when velocity code = 1 Delta x (dx) when velocity code = 0	Long	4	H+40	2^{-11}
13	1st half ddy	Delta delta y (ECEF) when velocity code = 1 Delta y (dy) when velocity code = 0	Long	4	H+44	2^{-11}

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
14	1st half ddz	Delta delta z (ECEF) when velocity code = 1 Delta z (dz) when velocity code = 0	Long	4	H+48	2^{-11}
15	1st half a_{f1}	Delta a_{f1} clock offset when velocity code = 1 Delta a_{f0} clock offset when velocity code = 0	Long	4	H+52	2^{-39}
16	1st half t_0	Applicable time of day Dummy value when velocity code = 0	Ulong	4	H+56	16
17	1st half iodp	Issue of PRN mask data	Ulong	4	H+60	-
18	1st half corr spare	Spare value when velocity code = 0 Dummy value when velocity code = 1	Ulong	4	H+64	-
19	2nd half vel	Velocity code flag (0 or 1)	Ulong	4	H+68	-
20	2nd half mask1	Index into PRN mask (Type 1)	Ulong	4	H+72	-
21	2nd half iode1	Issue of ephemeris data	Ulong	4	H+76	-
22	2nd half dx1	Delta x (ECEF)	Long	4	H+80	0.125
23	2nd half dy1	Delta y (ECEF)	Long	4	H+84	0.125
24	2nd half dz1	Delta z (ECEF)	Long	4	H+88	0.125
25	2nd half a_{f0}	Delta a_{f0} clock offset	Long	4	H+92	2^{-31}
26	2nd half mask2	Second index into PRN mask (Type 1) Dummy value when velocity code = 1	Ulong	4	H+96	-
27	2nd half iode2	Second issue of ephemeris data Dummy value when velocity code = 1	Ulong	4	H+100	-
28	2nd half ddx	Delta delta x (ECEF) when velocity code = 1 Delta x (dx) when velocity code = 0	Long	4	H+104	2^{-11}
29	2nd half ddy	Delta delta y (ECEF) when velocity code = 1 Delta y (dy) when velocity code = 0	Long	4	H+108	2^{-11}
30	2nd half ddz	Delta delta z (ECEF) when velocity code = 1 Delta z (dz) when velocity code = 0	Long	4	H+112	2^{-11}
31	2nd half a_{f1}	Delta a_{f1} clock offset when velocity code = 1 Delta a_{f0} clock offset when velocity code = 0	Long	4	H+116	2^{-39}
32	2nd half t_0	Applicable time of day Dummy value when velocity code = 0	Ulong	4	H+120	16
33	2nd half iodp	Issue of PRN mask data	Ulong	4	H+124	-
34	2nd half corr spare	Spare value when velocity code = 0 Dummy value when velocity code = 1	Ulong	4	H+128	-
35	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+132	-
36	[CR][LF]	Sentence terminator (ASCII only)	-	-	H+136	-

3.2.158 SBAS26
Ionospheric delay corrections

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS26 provides vertical delays (relative to an L1 signal) and their accuracy at geographically defined IGPs identified by the BAND NUMBER and IGP number. Each message contains a band number and a block ID, which indicates the location of the IGPs in the respective band mask.

Message ID: **985**
Log Type: **Asynch**

Recommended Input:

> log sbas26a onchanged

ASCII Example:

```
#SBAS26A,COM1,0,38.0,SATTIME,1337,417243.000,00000000,EC70,1984;
134,1,2,15,27,11,25,11,23,11,19,11,16,11,16,12,15,13,16,13,29,14,30
,13,27,11,27,11,24,11,19,11,
16,12,2,0*3B6D6806
```

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS26 message can be logged to view the data breakdown of SBAS frame 26 which contains ionospheric delay corrections

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS26 header	Log header		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	band num	Band number	Ulong	4	H+4	-
4	block id	Block ID	Ulong	4	H+8	-
5	#pts	Number of grid points with information to follow	Ulong	4	H+12	-
6	igp$_{vde}$	IGP vertical delay estimates	Ulong	4	H+16	0.125
7	givei	Grid ionospheric vertical error indicator	Ulong	4	H+20	-
8...	Next #pts entry = H + 16 + (#pts x 8)					
9	iodi	Issue of data - ionosphere	Ulong	4	H+16+ (#pts x 8)	
10	spare	7 spare bits	Ulong	4	H+20+ (#pts x 8)	-
11	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+24+ (#pts x 8)	-
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.159 SBAS27
SBAS service message

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS27 messages apply only to the service provider transmitting the message. The number of service messages indicates the total number of unique SBAS27 messages for the current IODS. Each unique message for that IODS includes a sequential message number. The IODS is incremented in all messages, each time that any parameter in any SBAS27 message is changed.

Message ID: 986
Log Type: Asynch

Recommended Input:

> log sbas27a onchanged

> (i) Each raw SBAS frame gives data for a specific frame decoder number. The SBAS27 message can be logged to view the data breakdown of SBAS frame 27 which contains information about SBAS service messages.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS27 header	Log header		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	iods	Issue of slow corrections data	Ulong	4	H+4	-
4	#messages	Low-by-one count of messages	Ulong	4	H+8	-
5	message num	Low-by-one message number	Ulong	4	H+12	-
6	priority code	Priority code	Ulong	4	H+16	-
7	dudre inside	Delta user differential range error - inside	Ulong	4	H+20	-
8	dudre outside	Delta user differential range error - outside	Ulong	4	H+24	-
9...	#reg	Number of regions with information to follow	Ulong	4	H+28	-
10	lat1	Coordinate 1 latitude	Long	4	H+32	-
11	lon1	Coordinate 1 longitude	Long	4	H+36	-
12	lat2	Coordinate 2 latitude	Long	4	H+40	-
13	lon2	Coordinate 2 longitude	Long	4	H+44	-
14	shape	Shape where: 0 = triangle 1 = square	Ulong	4	H+48	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
15	Next #reg entry = H+32+(#reg x 20)					
16	t_0	Time of applicability	Ulong	4	H+32+ (#reg x 20)	16
17	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+36+ (#reg x 20)	-
18	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.160 SBAS32
Fast correction slots 0-10

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS32 are fast corrections for slots 0-10 in the mask of SBAS1, see *page 678*.

Message ID: **988**
Log Type: **Asynch**

Recommended Input:

        log sbas32a onchanged

ASCII Example:

```
#SBAS32A,COM2,0,70.5,FINE,1295,153284.000,00000240,18E9,34461;209,
0,0,-8097,0,0,0,0,-947,0,-2128,0,2570,14,0,14,14,14,14,0,14,0,14,
0*58778AE5
```

Table 146: Evaluation of UDREI

UDREI	UDRE metres
0	0.01
1	0.02
2	0.03
3	0.05
4	0.10
5	0.15
6	0.20
7	0.25
8	0.30
9	0.35
10	0.40
11	0.45
12	0.50
13	0.60
14	Not Monitored
15	Do Not Use

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS32 header	Log header		H	0	
2	prn	Source PRN of message	Ulong	4	H	-
3	iodp	Issue of PRN mask data	Ulong	4	H+4	-
4	prc0		Long	4	H+8	-
5	prc1		Long	4	H+12	-
6	prc2		Long	4	H+16	-
7	prc3		Long	4	H+20	-
8	prc4	prc(i):	Long	4	H+24	-
9	prc5		Long	4	H+28	-
10	prc6	Fast corrections (-2048 to +2047) for the prn in slot i (i = 0-10)	Long	4	H+32	-
11	prc7		Long	4	H+36	-
12	prc8		Long	4	H+40	-
13	prc9		Long	4	H+44	-
14	prc10		Long	4	H+48	-
15	udre0		Ulong	4	H+52	
16	udre1		Ulong	4	H+56	
17	udre2		Ulong	4	H+60	
18	udre3		Ulong	4	H+64	
19	udre4	udre(i):	Ulong	4	H+68	See *Table 146, Evaluation of UDREI* on *page 708*
20	udre5		Ulong	4	H+72	
21	udre6	User differential range error indicator for the prn in slot i (i = 0-10)	Ulong	4	H+76	
22	udre7		Ulong	4	H+80	
23	udre8		Ulong	4	H+84	
24	udre9		Ulong	4	H+88	
25	udre10		Ulong	4	H+92	
26	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+96	-
27	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.161 SBAS33
Fast correction slots 11-21

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS33 are fast corrections for slots 11-21.

Message ID:　　　**989**

Log Type:　　　**Asynch**

Recommended Input:

　　　　log sbas33a onchanged

ASCII Example:
```
#SBAS33A,COM2,0,47.5,FINE,1295,158666.000,01000240,B23E,34461;209,
0,0,-3343,0,0,0,-533,0,0,0,0,0,14,0,14,14,14,0,14,14,14,14,
14*6D890F5F
```

> ⓘ Each raw mask frame gives data for a specific frame decoder number. The SBAS33 message can be logged to view the data breakdown of SBAS frame 33 which contains information about correction slots 11-21.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS33 header	Log header		H	0	
2	prn	Source PRN of message	Ulong	4	H	-
3	iodp	Issue of PRN mask data	Ulong	4	H+4	-
4	prc11		Long	4	H+8	-
5	prc12		Long	4	H+12	-
6	prc13		Long	4	H+16	-
7	prc14		Long	4	H+20	-
8	prc15	prc(i):	Long	4	H+24	-
9	prc16		Long	4	H+28	-
10	prc17	Fast corrections (-2048 to +2047) for the prn in slot i (i = 11-21)	Long	4	H+32	-
11	prc18		Long	4	H+36	-
12	prc19		Long	4	H+40	-
13	prc20		Long	4	H+44	-
14	prc21		Long	4	H+48	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
15	udre11		Ulong	4	H+52	
16	udre12		Ulong	4	H+56	
17	udre13		Ulong	4	H+60	
18	udre14		Ulong	4	H+64	
19	udre15	udre(i):	Ulong	4	H+68	See *Table 146, Evaluation of UDREI* on *page 708*
20	udre16	User differential range error indicator for the prn in slot i (i = 11-21)	Ulong	4	H+72	
21	udre17		Ulong	4	H+76	
22	udre18		Ulong	4	H+80	
23	udre19		Ulong	4	H+84	
24	udre20		Ulong	4	H+88	
25	udre21		Ulong	4	H+92	
26	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+96	-
27	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.162 SBAS34
Fast correction slots 22-32

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS34 are fast corrections for slots 22-32 in the mask of SBAS1, see *page 678*.

Message ID: **990**
Log Type: **Asynch**

Recommended Input:

> log sbas34a onchanged

ASCII Example:

```
#SBAS34A,COM2,0,73.0,FINE,1295,226542.000,00000040,1BE8,34461;209,
0,5879,0,0,0,0,2687,0,10922,10922,10922,10922,0,14,14,14,14,0,14,
15,15,15,15*3AEB74BE
```

> (i) Each raw mask frame gives data for a specific frame decoder number. The SBAS34 message can be logged to view the data breakdown of SBAS frame 34 which contains information about fast correction slots 22-32.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS34 header	Log header		H	0	
2	prn	Source PRN of message	Ulong	4	H	-
3	iodp	Issue of PRN mask data	Ulong	4	H+4	-
4	prc22		Long	4	H+8	-
5	prc23		Long	4	H+12	-
6	prc24		Long	4	H+16	-
7	prc25		Long	4	H+20	-
8	prc26	prc(i):	Long	4	H+24	-
9	prc27		Long	4	H+28	-
10	prc28	Fast corrections (-2048 to +2047) for the prn in slot i (i = 22-32)	Long	4	H+32	-
11	prc29		Long	4	H+36	-
12	prc30		Long	4	H+40	-
13	prc31		Long	4	H+44	-
14	prc32		Long	4	H+48	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
15	udre22	udre(i): User differential range error indicator for the prn in slot i (i = 22-32)	Ulong	4	H+52	See *Table 146, Evaluation of UDREI* on *page 708*
16	udre23		Ulong	4	H+56	
17	udre24		Ulong	4	H+60	
18	udre25		Ulong	4	H+64	
19	udre26		Ulong	4	H+68	
20	udre27		Ulong	4	H+72	
21	udre28		Ulong	4	H+76	
22	udre29		Ulong	4	H+80	
23	udre30		Ulong	4	H+84	
24	udre31		Ulong	4	H+88	
25	udre32		Ulong	4	H+92	
26	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+96	-
27	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.163 SBAS35
Fast correction slots 33-43

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

SBAS35 are fast corrections for slots 33-43 in the mask of SBAS1, see *page 678*.

Message ID: **991**
Log Type: **Asynch**

Recommended Input:

        log sbas35a onchanged

ASCII Example:

> ⓘ Each raw mask frame gives data for a specific frame decoder number. The SBAS35 message can be logged to view the data breakdown of SBAS frame 35 which contains information about fast correction slots 33-43.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS35 header	Log header		H	0	
2	prn	Source PRN of message	Ulong	4	H	-
3	iodp	Issue of PRN mask data	Ulong	4	H+4	-
4	prc33		Long	4	H+8	-
5	prc34		Long	4	H+12	-
6	prc35		Long	4	H+16	-
7	prc36		Long	4	H+20	-
8	prc37	prc(i):	Long	4	H+24	-
9	prc38		Long	4	H+28	-
10	prc39	Fast corrections (-2048 to +2047) for the prn in slot i (i = 33-43)	Long	4	H+32	-
11	prc40		Long	4	H+36	-
12	prc41		Long	4	H+40	-
13	prc42		Long	4	H+44	-
14	prc43		Long	4	H+48	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
15	udre33	udre(i): User differential range error indicator for the prn in slot i (i = 33-43)	Ulong	4	H+52	See *Table 146, Evaluation of UDREI* on *page 708*
16	udre34		Ulong	4	H+56	
17	udre35		Ulong	4	H+60	
18	udre36		Ulong	4	H+64	
19	udre37		Ulong	4	H+68	
20	udre38		Ulong	4	H+72	
21	udre39		Ulong	4	H+76	
22	udre40		Ulong	4	H+80	
23	udre41		Ulong	4	H+84	
24	udre42		Ulong	4	H+88	
25	udre43		Ulong	4	H+92	
26	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+96	-
27	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.164 SBAS45
Slow corrections

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

Each SBAS45 message contains a 2-bit IODP indicating the associated PRN mask.

The time of applicability (T0) of the PRC is the start of the epoch of the WNT second that is coincident with the transmission at the satellite of the first bit of the message block.

Message ID: **993**
Log Type: **Asynch**

Recommended Input:
```
log SBAS45a onchanged
```

ASCII Example:
```
#SBAS45A,COM2,0,73.0,FINE,1295,228498.000,00000040,C730,34461;209,
23,32,197,-116,206,-1,-6,-3,-5546,3488,25,148,262,-312,867,4,3,0,
2513,3488,0*02D6E0D5
```

> (i) Each raw mask frame gives data for a specific frame decoder number. The SBAS45 message can be logged to view the data breakdown of SBAS frame 45 which contains information about slow corrections.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
1	SBAS45 header	Log header		H	0	-
2	prn	Source PRN of message	Ulong	4	H	-
3	mask1	Index into PRN mask (Type 1)	Ulong	4	H+4	-
4	iode1	Issue of ephemeris data	Ulong	4	H+8	-
5	dx1	Delta x (ECEF)	Long	4	H+12	0.125
6	dy1	Delta y (ECEF)	Long	4	H+16	0.125
7	dz1	Delta z (ECEF)	Long	4	H+20	0.125
8	ddx1	Delta delta x (ECEF)	Long	4	H+24	2^{-11}
9	ddy1	Delta delta y (ECEF)	Long	4	H+28	2^{-11}
10	ddz1	Delta delta z (ECEF)	Long	4	H+32	2^{-11}
11	$da_{f0}1$	Delta a_{f0} clock offset	Long	4	H+36	2^{-31}
12	t_01	Applicable time of day	Ulong	4	H+40	16
13	mask2	Second index into PRN mask (Type 1)	Ulong	4	H+44	-
14	iode2	Second issue of ephemeris data	Ulong	4	H+48	-

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset	Scaling
15	dx2	Delta x (ECEF)	Long	4	H+52	0.125
16	dy2	Delta y (ECEF)	Long	4	H+56	0.125
17	dz2	Delta z (ECEF)	Long	4	H+60	0.125
18	ddx2	Delta delta x (ECEF)	Long	4	H+64	2^{-11}
19	ddy2	Delta delta y (ECEF)	Long	4	H+68	2^{-11}
20	ddz2	Delta delta z (ECEF)	Long	4	H+72	2^{-11}
21	$da_{f0}2$	Delta a_{f0} clock offset	Long	4	H+76	2^{-31}
22	$t_0 2$	Applicable time of day	Ulong	4	H+80	16
23	iodp	Issue of PRN mask data	Ulong	4	H+84	-
24	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+88	-
25	[CR][LF]	Sentence terminator (ASCII only)	-	-	-	-

3.2.165 SBASALMANAC
SBAS Almanac collection

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains a collection of all current SBAS almanacs that have been decoded by the receiver and may contain almanac data for multiple PRNs. The SBASALMANAC log is populated by the Geo Almanac Message Type 17 and is also available in the SBAS17 log. These PRNs are broken out into individual SBAS almanac entries for this message and output individually. If multiple SBAS subsystems (e.g., WAAS, EGNOS, GAGAN, MSAS) are tracked, this message will include almanac data collected from each with the subsystem identified in each message entry. The almanac data contains all of the information required to compute the satellite position as well as health and status information.

Message ID:	**1425**
Log Type:	**Asynch**

Recommended Input:

> log sbasalmanaca onchanged

ASCII Example:

```
#SBASALMANACA,COM1,2,80.0,SATTIME,1672,411186.000,00000020,84D8,
43119;133,WAAS,65600,0,0,-5571800,-41758600,-1456000,0,0,120
*22DA17E8
#SBASALMANACA,COM1,1,80.0,SATTIME,1672,411186.000,00000020,84D8,
43119;135,WAAS,65600,0,0,-28758600,-30825600,0,0,0,0*DD122CA1
#SBASALMANACA,COM1,0,80.0,SATTIME,1672,411186.000,00000020,84D8,
43119;138,WAAS,65600,0,0,-12547600,-40248000,0,0,0,0*89C6C51C
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Header	Log header		H	0
2	ID	Satellite ID	Ulong	4	H
3	Variant	System variant (refer to *Table 147, SBAS Subsystem Types* on *page 719*)	Enum	4	H + 4
4	Time	Time of day (s)	Ulong	4	H + 8
5	Data ID	Data identification	Ushort	2	H + 12
6	Health	Satellite health	Ushort	2	H + 14
7	X	ECEF X coordinate (m)	Long	4	H + 16
8	Y	ECEF Y coordinate (m)	Long	4	H + 20
9	Z	ECEF Z coordinate (m)	Long	4	H + 24
10	X Velocity	X rate of change (m/s)	Long	4	H + 28
11	Y Velocity	Y rate of change (m/s)	Long	4	H + 32

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
12	Z Velocity	Z rate of change (m/s)	Long	4	H + 36
13	CRC	32-bit CRC (ASCII and binary only)	Ulong	4	H + 40
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 147: SBAS Subsystem Types

ASCII	Binary	Description
NONE	0	No system
UNKNOWN	1	Unknown system
WAAS	2	WAAS system
EGNOS	3	EGNOS system
MSAS	4	MSAS system
GAGAN	5	GAGAN system
QZSS	7	QZSS System

3.2.166 SBASCORR
SBAS range corrections used

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The information is updated with each pseudorange position calculation. It has an entry for each tracked satellite. Satellites that are not included in an SBAS corrected solution have 0.0 in both the correction and 'corr stdv' fields.

The correction is the combined fast and slow corrections and is to be added to the pseudorange. Ionospheric and tropospheric corrections are not included and should be applied separately.

Message ID: **998**
Log Type: **Synch**

Recommended Input:

      log sbascorra ontime 1

ASCII Example:

```
#SBASCORRA,COM1,0,40.5,FINESTEERING,1337,417485.000,01000000,3B3B,
1984;
20,
3,101,0.0000,0.0000,3,0,0.0000,0.0000,
2,133,0.0000,0.0000,2,0,0.0000,0.0000,
23,48,0.0000,0.0000,23,0,0.0000,0.0000,
4,55,0.0000,0.0000,4,0,0.0000,0.0000,
16,197,0.0000,0.0000,16,0,0.0000,0.0000,
20,25,0.0000,0.0000,20,0,0.0000,0.0000,
27,26,0.0000,0.0000,27,0,0.0000,0.0000,
25,186,0.0000,0.0000,25,0,0.0000,0.0000,
13,85,0.0000,0.0000,13,0,0.0000,0.0000,
122,0,0.0000,0.0000,134,0,0.0000,0.0000*0AF4C14D
```

> ⓘ The SBAS corrections are applied to the raw pseudorange measurements which results in a more accurate solution, without the user requiring any additional hardware.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	SBASCORR header	Log header		H	0
2	#sat	Number of satellites with information to follow	Ulong	4	H
3	prn	Satellite PRN	Ulong	4	H+4

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
4	iode	Issue of ephemeris data for which the corrections apply	Ulong	4	H+8
5	corr	SBAS pseudorange correction (m)	Float	4	H+12
6	corr stdv	Standard deviation of pseudorange correction (m)	Float	4	H+16
7...	Next sat entry = H+4+(#sat x 16)				
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#sat x 16)
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.167 SOFTLOADSTATUS
Describes the status of the SoftLoad process

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log describes the status of the SoftLoad process.

> ⓘ Status values >= 16 (ERROR) indicate that an error has occurred during the loading process. Status < 16 (ERROR) are part of normal SoftLoad operation.

Message ID: **1235**
Log Type: **Asynch**

Recommended Input:

> log softloadstatusa onchanged

ASCII Example:

```
#SOFTLOADSTATUSA,COM1,0,97.5,UNKNOWN,0,0.113,004C0001,2D64,10481;
NOT_STARTED*827FDC04
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	SOFTLOADSTATUS header	Log header	-	H	0
2	status	Status of the SoftLoad process see the following table	Enum	4	H
3	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	
4	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Table 148: SoftLoad Status Type

Value	Name	Description
1	NOT_STARTED	SoftLoad process has not begun
2	READY_FOR_SETUP	SoftLoad process is ready to receive setup information in the form of SOFTLOADSETUP commands or SOFTLOADSREC commands with S0 records.Once sufficient setup data has been sent, the process is also ready for SOFTLOADDATA commands
3	READY_FOR_DATA	SoftLoad process is ready to receive data in the form of SOFTLOADDATA commands or SOFTLOADSREC commands with S3 records. Once all data has been sent, send the SOFTLOADCOMMIT command
4	DATA_VERIFIED	SoftLoad data has passed CRC. This status occurs after a SOFTLOADCOMMIT command
5	WRITING_FLASH	SoftLoad data is being written to flash. This status occurs after a SOFTLOADCOMMIT command. During a firmware upload, the receiver may remain in this state for 300 seconds or longer

Value	Name	Description
6	WROTE_FLASH	SoftLoad data has been written to flash
7	WROTE_AUTHCODE	The embedded AuthCode was successfully written
8	COMPLETE	SoftLoad process has completed. The next step is to send the RESET command to reset the receiver
9	VERIFYING_DATA	SoftLoad is verifying the downloaded image
10	COPIED_SIGNATURE_AUTH	Signature AuthCodes have been copied from the current firmware to the downloaded firmware.
11	WROTE_TRANSACTION_TABLE	The downloaded firmware has been activated and will be executed if the receiver is reset. This status is effectively identical to COMPLETE.
16	ERROR	Indicates an internal error in the SoftLoad process. This error is not expected to occur. Contact NovAtel Customer Support for assistance.
17	RESET_ERROR	Error reseting SoftLoad. Reset the receiver and restart the SoftLoad process.
18	BAD_SRECORD	A bad S Record was received. Ensure that S Records are enclosed in double quotes within the SOFTLOADSREC command.
19	BAD_PLATFORM	This data cannot be loaded onto this platform. Ensure that the correct *.hex or *.shex file for the platform is being used.
20	BAD_MODULE	This module cannot be loaded with SoftLoad. This file must be loaded using WinLoad or a similar loader.
21	BAD_AUTHCODE	Bad AuthCode received for this PSN
22	NOT_READY_FOR_SETUP	A SOFTLOADSETUP command was entered before a SOFTLOADRESET or after a SOFTLOADDATA command
23	NO_MODULE	No data type was entered before a SOFTLOADDATA command was received. Set the data type using the SOFTLOADSETUP command or SOFTLOADSREC command with an "S0~T~" S Record.
24	NO_PLATFORM	No platform was entered before a SOFTLOADDATA command was received. Set the platform using the SOFTLOADSETUP command or SOFTLOADSREC command with an "S0~P~" S Record.
25	NOT_READY_FOR_DATA	A SOFTLOADDATA command was received but the receiver was not ready for it
26	MODULE_MISMATCH	The SoftLoad data module was changed in the middle of loading. Restart the SoftLoad process using the SOFTLOADRESET command.
27	OUT_OF_MEMORY	SoftLoad has run out of RAM to store the incoming data. Reset the receiver and restart the SoftLoad process.
28	DATA_OVERLAP	SoftLoad data has overlapped. Ensure that the correct address and length is set in the SOFTLOADDATA or SOFTLOADSREC command.

Value	Name	Description
29	BAD_IMAGE_CRC	CRC of the downloaded image has failed. Ensure that all content from the *.hex or *.shex file has been successfully downloaded.
30	IMAGE_OVERSIZE	The downloaded image is too big for the intended data module
31	AUTHCODE_WRITE_ERROR	An error occurred when writing the embedded AuthCode to flash
32	BAD_FLASH_ERASE	Erasing of the flash failed. This could indicate a failure in the flash hardware.
33	BAD_FLASH_WRITE	Writing to the flash failed. This could indicate a failure in the flash hardware.
34	TIMEOUT	SoftLoad time out has occurred

3.2.168 SOURCETABLE
NTRIP source table entries

OEM Platform: 628, 638, FlexPak6, ProPak6

This log outputs the NTRIP SOURCETABLE entries from the NTRIPCASTER set by the NTRIPSOURCETABLE command (*page 224*). The entry data field in the first entry is always the header of the retrieved SOURCETABLE. The entry data field in the last entry is always a string "ENDSOURCETABLE" which indicates the end of the source table. Entries in between these fields are the real SOURCETABLE entries.

Message ID: **1344**
Log Type: **Synch**

Recommended Input:

            log sourcetablea once

ASCII Example:

```
#SOURCETABLEA,ICOM1,7,70.0,FINESTEERING,1642,150600.299,00000020,
275D,6883;"198.161.64.11:80","HTTP/1.1 200 OK;NTRIP-VERSION: NTRIP/
2.0;NTRIP-FLAGS: ST_FILTER,ST_AUTH,ST_MATCH,ST_STRICT,RTSP,
PLAIN_RTP;SERVER: NTRIP CASTER/2.0.15;DATE: MON, 27 JUN 2011
17:47:23 GMT;CONNECTION: CLOSE;CONTENT-TYPE: GNSS/SOURCETABLE;
CONTENT-LENGTH: 671"*6D385807

#SOURCETABLEA,ICOM1,6,70.0,FINESTEERING,1642,150600.299,00000020,
275D,6883;"198.161.64.11:80","CAS;HERA.NOVATEL.CA;80,2101;NOVATEL;
NOVATEL;0;CAN;51;-115;HTTP://WWW.NOVATEL.COM"*D5DCF61B

#SOURCETABLEA,ICOM1,5,72.0,FINESTEERING,1642,150600.299,00000020,
275D,6883;"198.161.64.11:80","NET;GREF;NOVATEL;B;N;HTTP://
NOVATEL.COM;NONE;NOVATEL.COM;NONE"*E1ABE7EF

#SOURCETABLEA,ICOM1,4,72.0,FINESTEERING,1642,150600.299,00000020,
275D,6883;"198.161.64.11:80","STR;NOVATEL_RTCMV3;NOVATEL RTCMV3;
RTCM 3.0;1004(1),1006(1),1012(1),1033(1);2;GPS+GLO;NOVATEL;CAN;51;
-15;0;1;NOVATEL OEM628;NONE;B;N;9600;TEST"*59CD860F

#SOURCETABLEA,ICOM1,3,72.0,FINESTEERING,1642,150600.299,00000020,
275D,6883;"198.161.64.11:80","STR;NOVATEL_RTCM;NOVATEL RTCM;RTCM;
1(1),3(10),31(1),32(10);2;GPS+GLO;NOVATEL;CAN;51;-15;0;1;NOVATEL
OEM628;NONE;B;N;9600;TEST"*3A8DC2FF

#SOURCETABLEA,ICOM1,2,72.0,FINESTEERING,1642,150600.299,00000020,
275D,6883;"198.161.64.11:80","STR;NOVATEL_RTCAOBS2;NOVATEL RT-
CAOBS2;RTCA; RTCAREF(10), RTCAOBS2(1);2;GPS+GLO;NOVATEL;CAN;51;
-15;0;1;NOVATEL OEM628;NONE;B;N;9600;TEST"*7078FA36

#SOURCETABLEA,ICOM1,1,72.0,FINESTEERING,1642,150600.299,00000020,
275D,6883;"198.161.64.11:80","STR;IPG;IPG TEST;UNKNOWN;UNKNOWN;2;
UNKNOWN;NOVATEL;CAN;51;-115;0;1;NOVATEL OEM628;NONE;B;N;1200;
TEST"*E0F2CF39

#SOURCETABLEA,ICOM1,0,72.0,FINESTEERING,1642,150600.299,00000020,
275D,6883;"198.161.64.11:80","ENDSOURCETABLE"*2C5015C9
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	SOURCETABLE header	Log header		H	0
2	endpoint	NTRIPCASTER Endpoint	String with varied length up to 80 bytes	a*	H
3	Reserved1	reserved	Ulong	4	H+a
4	Reserved2	reserved	Ulong	4	H+a+4
5	Entry data	Source table entry data	String with varied length up to 512 bytes	b*	H+a+8
6	xxxx	32-bit CRC (ASCII and binary only)	Ulong	4	H+a+b+8
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

*In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

3.2.169 *TERRASTARINFO*
TerraStar subscription information

OEM Platform: 628, 638, FlexPak6, ProPak6

This log contains details on the TerraStar subscription.

Message ID: **1719**
Log Type: **Asynch**

Recommended Input:

        log terrastarinfoa onchanged

ASCII Example:
```
#TERRASTARINFOA,COM1,0,65.5,UNKNOWN,
0,1.168,00040008,E776,13260;"QR391:3006:6179",TERM,00000301,0,0
,0,NONE,0.00000,0.00000,0*7E4A9EC0
```

Table 149: TerraStar Subscription Details Mask

Bit	Mask	Description
8-9	0x3	1 = TerraStar-D subscription 2 = TerraStar-C subscription

Table 150: TerraStar Subscription Type

ASCII	Binary	Description
UNASSIGNED	0	Decoder has not had an assigned operating mode
TERM	1	Term subscription
PAYG	2	Pay-As-You-Go (PAYG) subscription
BUBBLE	100	Receiver is operating in a TerraStar-permitted subscription-free bubble
MODEL_DENIED	101	TerraStar is not permitted on the current firmware model

Table 151: TerrStar Region Restriction

ASCII	Binary	Description
NONE	0	TerraStar operation is not permitted
GEOGATED	1	TerraStar operation limited to on-land
LOCAL_AREA	2	TerraStar operation limited to radius from local area center point
NEARSHORE	3	TerraStar operation limited to on land and nearshort (coastal) regions

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	TERRA STARINFO header	Log header		H	0
2	PAC	Product activation code	Char[16]	16	H
3	Type	Subscription type (see *Table 150, TerraStar Subscription Type* above)	Enum	4	H+16
4	Details	Subscription details (see *Table 149, TerraStar Subscription Details Mask* above)	Hex	4	H+20
5	Contract end year	Year that subscription ends	Ulong	4	H+24
6	Contract end DOY	Day-of-year (DOY) that subscription ends	Ulong	4	H+28
7	Enable period	For PAYG subscriptions, the duration of the subscription (hours)	Ulong	4	H+32
8	Region restriction	For region restricted subscriptions, the type of region restriction (see *Table 151, TerrStar Region Restriction*)	Enum	4	H+36
9	Center point latitude	For local area subscriptions, the centre point latitude (degrees)	Float	4	H+40
10	Center point longitude	For local area subscriptions, the centre point longitude (degrees)	Float	4	H+44
11	Radius	For local area subscriptions, the maximum permitted distance from centre point (kilometers)	Ulong	4	H+48
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+52
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.170 TERRASTARSTATUS
TerraStar decoder and subscription status

OEM Platform: 628, 638, FlexPak6, ProPak6

This log contains status information for the TerraStar decoder and subscription.

Message ID: **1729**
Log Type: **Asynch**

Recommended Input:

       log terrastarstatusa onchanged

ASCII Example:
```
#TERRASTARSTATUSA,COM1,0,49.5,FINESTEERING,
1769,332336.443,00000000, FDC1,12602;ENABLE,LOCKED,0,DISABLED
ONSHORE*555155A5
```

Table 152: TerraStar Decoder Data Synchronization State

ASCII	Binary	Description
NO_SIGNAL	0	Decoder has not received L-band for more than 30 seconds
SEARCH	1	Decoder is searching for format
LOCKED	2	Decoder is locked to the data format

Table 153: TerraStar Local Area Status

ASCII	Binary	Description
DISABLED	0	Waiting for or unknown local area status
WAITING_FOR_POSITION	1	Waiting for a position
RANGE_CHECK	16	Checking position against local area region restriction
IN_RANGE	129	Receiver is within the permitted local area
OUT_OF_RANGE	130	Receiver is outside the permitted local area
POSITION_TOO_OLD	255	Position is too old

Table 154: TerraStar Geogating Status

ASCII	Binary	Description
DISABLED	0	Waiting for or unknown geogating area status
WAITING_FOR_POSITION	1	Waiting for a position
ONSHORE	129	Receiver is over land
OFFSHORE	130	Receiver is over water
POSITION_TOO_OLD	255	Position is too old
PROCESSING	1000	Geogater is determining status

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	TERRA STAR STATUS header	Log header		H	0
2	Access	Access status. ENABLE (1) if the subscription is valid; DISABLE (0) otherwise	Enum	4	H
3	Sync state	Decoder data synchronization state (see *Table 152, TerraStar Decoder Data Synchronization State* above)	Enum	4	H+4
4	Timed enable remaining time	For PAYG subscriptions, minutes remaining on subscription.	Ulong	4	H+8
5	Local area status	For local-area subscriptions, indicates if the receiver is within the permitted area (see *Table 153, TerraStar Local Area Status* above)	Enum	4	H+12
6	Geogating status	Geogating status (see *Table 154, TerraStar Geogating Status* above)	Enum	4	H+16
7	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+20
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.171 TIME
Time data

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log provides several time related pieces of information including receiver clock offset and UTC time and offset. It can also be used to determine any offset in the PPS signal relative to GPS reference time.

To find any offset in the PPS signal, log the TIME log 'ontime' at the same rate as the PPS output. For example, if the PPS output is configured to output at a rate of 0.5 seconds, see the `PPSCONTROL` command on *page 239*, log the TIME log 'ontime 0.5' as follows:

```
log time ontime 0.5
```

The TIME log offset field can then be used to determine any offset in PPS output relative to GPS reference time.

Message ID: **101**
Log Type: **Synch**

Recommended Input:

> log timea ontime 1

ASCII Example:

```
#TIMEA,COM1,0,50.5,FINESTEERING,1337,410010.000,00000000,9924,1984;
VALID,1.953377165E-09,7.481712815E-08,-12.99999999492,2005,8,25,17,
53,17000,VALID*E2FC088C
```

ⓘ 1. Consider the case where you used the ADJUST1PPS command, see *page 61*, to synchronize two receivers in a primary/secondary relationship to a common external clock. You can use the TIME log after the clock model has stabilized at state 0, to monitor the time difference between the Primary and Secondary receivers.

    2. The header of the TIME log gives you the GPS reference time (the week number since January 5th, 1980) and the seconds into that week. The TIME log outputs the UTC offset (offset of GPS reference time from UTC time) and the receiver clock off-set from GPS reference time.

If you want the UTC time in weeks and seconds, take the week number from the header. Then take the seconds into that week, also from the header, and add the correction to the seconds using the 2 offsets. Ensure not to go negative or rollover (go over the total number of seconds, 604800, in a week). In the case of a rollover, add a week and the left over seconds become the seconds into this new week. If negative, subtract a week and the remainder from the seconds of that week become the seconds into this new week.

For example:

TIME COM1 0 73.5 FINESTEERING 1432 235661.000 00000000 9924 2616

VALID -0.000000351 0.000000214 -14.00000000106 2007 6 19 17 27 27000 VALID

From the time information above:

GPS reference time = 1432 (GPS reference week), 235661.000 (GPS seconds) from the header.

From the UTC offset row in the TIME log description on *page 733*:

UTC time = GPS reference time + offset + UTC offset

UTC time

= week 1432, 235661.000 s - 0.000000351 (offset) - 14.00000000106 (UTC offset)
= week 1432, seconds 235646.99999964794

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	TIME header	Log header		H	0
2	clock status	Clock model status (not including current measurement data), see *Table 99* on *page 433*	Enum	4	H
3	offset	Receiver clock offset, in seconds from GPS reference time. A positive offset implies that the receiver clock is ahead of GPS reference time. To derive GPS reference time, use the following formula: GPS reference time = receiver time - offset	Double	8	H+4
4	offset std	Receiver clock offset standard deviation (s)	Double	8	H+12

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
5	utc offset	The offset of GPS reference time from UTC time, computed using almanac parameters. UTC time is GPS reference time plus the current UTC offset plus the receiver clock offset: UTC time = GPS reference time + offset + UTC offset	Double	8	H+20
6	utc year	UTC year	Ulong	4	H+28
7	utc month	UTC month (0-12) [a]	Uchar	1	H+32
8	utc day	UTC day (0-31) [a]	Uchar	1	H+33
9	utc hour	UTC hour (0-23)	Uchar	1	H+34
10	utc min	UTC minute (0-59)	Uchar	1	H+35
11	utc ms	UTC millisecond (0-60999) [b]	Ulong	4	H+36

12	utc status	UTC status
		      0 = Invalid
		      1 = Valid
		      2 = Warning[c]
			Enum	4	H+40

13	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+44
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. If UTC time is unknown, the values for month and day are 0.

b. Maximum of 60999 when leap second is applied.

c. Indicates that the leap seconds value is used as a default due to the lack of an almanac.

3.2.172 TIMESYNC
Synchronize time between GNSS receivers

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The TIMESYNC log is used in conjunction with the ADJUST1PPS command, see *page 61*, to synchronize the time between GNSS receivers.

Message ID: **492**
Log Type: **Synch**

Recommended Input:

   log timesynca ontime 1

ASCII Example:

```
#TIMESYNCA,COM1,0,46.0,FINESTEERING,1337,410095.000,00000000,BD3F,
1984;1337,410095000,FINESTEERING*AA2025DB
```

> (i) The time data embedded in this log represents the time of the most recent 1PPS signal. The receiver issues this log from a communications port within 200 ms of the last 1PPS event. The 200 ms value is a "worst case scenario." Refer to Figure 2, 1PPS Alignment on *page 61* to see the alignment between a Fine and a Cold Clock receiver. Also refer to the *Transfer Time Between Receivers* section in the OEM6 Family Installation and Operation User Manual (OM-20000128).

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	TIMESYNC header	Log header		H	0
2	week	GPS reference week number	Ulong	4	H
3	ms	Number of milliseconds into the GPS reference week	Ulong	4	H+4
4	time status	GPS reference time Status, see *Table 8, GPS Reference Time Status* on *page 35*	Enum	4	H+8
5	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+12
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.173 TRACKSTAT
Tracking status

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

The TRACKSTAT log contains an entry for each channel. If there are multiple signal channels for one satellite (for example L1, L2 P(Y), L2C, and L5 for GPS), then there will be multiple entries for that satellite. The signal type can be determined from the channel tracking status word.

If both the L1 and L2 signals are being tracked for a given PRN, two entries with the same PRN appear in the tracking status log. As shown in *Table 126, Channel Tracking Status* on *page 604* these entries can be differentiated by bit 20, which is set if there are multiple observables for a given PRN, and bits 21-25, which denote the signal type for the observation. This is to aid in parsing the data.

A zero in the PRN/slot of the TRACKSTAT log indicates the channel should be considered idle with the exception of those for GLONASS. A GLONASS channel should only be considered idle if the tracking state is 0 in the channel tracking status word.

Message ID: **83**
Log Type: **Synch**

Recommended Input:

        log trackstata ontime 1

ASCII Example:

```
#TRACKSTATA,COM1,0,49.5,FINESTEERING,1337,410139.000,00000000,457C,
1984;SOL_COMPUTED,PSRDIFF,5.0,30,
1,0,18109C04,21836080.582,-2241.711,50.087,1158.652,0.722,GOOD,0.973,
1,0,11309C0B,21836083.168,-1746.788,42.616,1141.780,0.000,OBSL2,0.000,
30,0,18109C24,24248449.644,-2588.133,45.237,939.380,-0.493,GOOD,0.519,
30,0,11309C2B,24248452.842,-2016.730,38.934,939.370,0.000,OBSL2,0.000,
...
14,0,18109DA4,24747286.206,-3236.906,46.650,1121.760,-0.609,GOOD,0.514,
14,0,11309DAB,24747288.764,-2522.270,35.557,1116.380,0.000,OBSL2,0.000,
0,0,0C0221C0,0.000,0.000,0.047,0.000,0.000,NA,0.000,
0,0,0C0221E0,0.000,0.000,0.047,0.000,0.000,NA,0.000*255A732E
```

617D **FPk6D** For the OEM617D and FlexPak6D receivers, a TRACKSTAT_1 log can be requested to get TRACKSTAT data from the second antenna. As described in *Table 3, Binary Message Header Structure* on *page 26*, the message type indicates the log is from the second antenna. To request an ASCII log enter TRACKSTATA_1 and for a binary log enter TRACKSTATB_1.

> (i) For single point positioning, a minimum of 4 GPS satellites is required.
>
> For RTK and OmniSTAR HP/XP/G2, a minimum of 5 GPS satellites are required.
>
> Extra satellites provide additional redundancy, which is good to have. Note that the default cut-off angle is 5 degrees and single point positioning utilizes all available GPS satellites in the position solution.
>
> RTK solutions only use GPS satellites that are above the RTK elevation angle (usually 12.5 degrees). So, although there could be more than 5 GPS satellites in view, if there are not at least 5 GPS satellites above 12.5 degrees then an RTK solution may not be possible.

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	TRACKSTAT header	Log header		H	0
2	sol status	Solution status (see *Table 85, Solution Status* on *page 408*)	Enum	4	H
3	pos type	Position type (see *Table 84, Position or Velocity Type* on *page 407*)	Enum	4	H+4
4	cutoff	GPS tracking elevation cut-off angle	Float	4	H+8
5	# chans	Number of hardware channels with information to follow	Ulong	4	H+12
6	PRN/slot	Satellite PRN number of range measurement GPS: 1 to 32, SBAS: 120 to 138 or 183 to187, QZSS: 193-197, Galileo: 1 to 36, BDS: 1-30. For GLONASS, see *Section 1.3 on page 34*	Short	2	H+16
7	glofreq	(GLONASS Frequency + 7), see *Section 1.3* on *page 34*	Short	2	H+18
8	ch-tr-status	Channel tracking status (see *Table 126, Channel Tracking Status* on *page 604*)	ULong	4	H+20
9	psr	Pseudorange (m) - if this field is zero but the channel tracking status in the previous field indicates that the card is phase locked and code locked, the pseudorange has not been calculated yet	Double	8	H+24
10	Doppler	Doppler frequency (Hz)	Float	4	H+32
11	C/No	Carrier to noise density ratio (dB-Hz)	Float	4	H+36
12	locktime	Number of seconds of continuous tracking (no cycle slips)	Float	4	H+40

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
13	psr res	Pseudorange residual from pseudorange filter (m)	Float	4	H+44
14	reject	Range reject code from pseudorange filter (see *Table 90, Observation Statuses* on *page 413*)	Enum	4	H+48
15	psr weight	Pseudorange filter weighting	Float	4	H+52
16...	Next PRN offset = H+16+(#chans x 40)				
17	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+16+ (#chans x 40)
18	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.174 VALIDMODELS
Valid model information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log gives a list of valid authorized models available and expiry date information.

If a model has no expiry date, it reports the year, month and day fields as 0, 0 and 0 respectively.

Message ID: **206**
Log Type: **Asynch**

Recommended Input:

> log validmodelsa once

ASCII Example:

```
#VALIDMODELSA,COM1,0,92.0,FINESTEERING,1610,499139.682,00000000,
342F,6293;1,"D2LR0RCCR",0,0,0*D0580C1B
```

> ⓘ Use the VALIDMODELS log to output a list of available models for the receiver. Use the AUTH command, see *page 82*, to add a model and the MODEL command, see *page 216*, to change the currently active model. See the VERSION log on *page 742* for the currently active model

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	VALIDMODELS header	Log header		H	0
2	#mod	Number of models with information to follow	Ulong	4	H
3	model	Model name	String [Max16]	Variable[a]	Variable
4	expyear	Expiry year	Ulong	4	Variable Max:H+20
5	expmonth	Expiry month	Ulong	4	Variable Max: H+24
6	expday	Expiry day	Ulong	4	Variable: Max: H+28
7...	Next model offset = H+4+(#mods x variable [max:28])				
8	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#mods x variable [max:28])
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

   a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

3.2.175 VERIPOSINFO
Veripos subscription information

OEM Platform: 628, 617, 638, FlexPak6, ProPak6

This log contains details on the Veripos subscription.

Message ID: **1728**
Log Type: **Asynch**

Recommended Input:

        log veriposinfoa onchanged

ASCII Example:
```
#VERIPOSINFOA,COM2,0,60.5,FINESTEERING,
1779,176287.725,00044008,31FA,12740;320325,NCC_CON-
TROLLED,00000101,"Q"*26A9F04E
```

Table 155: Veripos Operating Mode

ASCII	Binary	Description
UNASSIGNED	0	Decoder has not had an assigned operating mode
NCC_CONTROLLED	7	Decoder operation disabled by a command from the Network Control Centre (NCC)
NO_DISABLE	8	Decoder operation not disabled
BUBBLE	100	Decoder is operating in a Veripos permitted subscription-free bubble
MODEL_DENIED	101	Decoder operation is not permitted on the current firmware model

Table 156: Veripos Subscription Details Mask

Bit	Mask	Description
0	0x001	Subscription permits differential positioning
8	0x100	Subscription permits Apex PPP positioning

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	VERIPOS INFO header	Log header		H	0
2	Serial number	Receiver serial number	Ulong	4	H
3	Mode	Operating mode (see *Table 155, Veripos Operating Mode* above)	Enum	4	H+4

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
4	Details	Subscription details (refer to *Table 156, Veripos Subscription Details Mask* on *page 739*)	Hex	4	H+12
5	Service code	Veripos service code	Char[4]	4	H+8
6	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+6
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.176 *VERIPOSSTATUS*
Veripos decoder and subscription status

OEM Platform: 628, 638, FlexPak6, ProPak6

This log contains status information for the Veripos decoder and subscription.

Message ID: **1730**
Log Type: **Asynch**

Recommended Input:

> log veriposstatusa onchanged

ASCII Example:

```
#VERIPOSSTATUSA,COM2,0,62.0,FINESTEERING,
1779,176955.656,00004008,0719,12740;ENABLE,LOCKED*7C5F85AE
```

Table 157: Veripos Decoder Data Synchronization State

ASCII	Binary	Description
NO_SIGNAL	0	Decoder has not received L-band for more than 30 seconds
SEARCH	1	Decoder is searching for format
LOCKED	2	Decoder is locked to the data format

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	VERIPOS STATUS header	Log header		H	0
2	Access	Access status. ENABLE (1) if the subscription is valid; DISABLE (0) otherwise	Enum	4	H
3	Sync state	Decoder data synchronization state (see *Table 157, Veripos Decoder Data Synchronization State* above)	Enum	4	H+4
4	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+8
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.177 VERSION
Version information

OEM Platform: 615, 617, 617D, 628, 638, FlexPak6, FlexPak6D, ProPak6

This log contains the version information for all components of a system. When using a standard receiver, there is only one component in the log.

A component may be hardware (for example, a receiver or data collector) or firmware in the form of applications or data (for example, data blocks for height models or user applications). See *Table 159, VERSION Log Field Formats* on *page 743* for details on the format of key fields.

See also the VALIDMODELS log on *page 738*.

Message ID: **37**
Log Type: **Polled**

Recommended Input:

> log versiona once

ASCII Example:

```
#VERSIONA,COM1,0,89.0,FINESTEERING,1610,504872.194,00000000,3681,
6293;1,GPSCARD,"D2LR0RCCR","BFN10260022","OEM628-.00",
"OEM060000RN0000","OEM060000SB0002","2010/NOV/05","16:11:18"*
```

> ⓘ The VERSION log is a useful log as a first communication with your receiver. Once connected, using NovAtel's Connect or HyperTerminal, log VERSION and check that the output makes sense. Also, ensure that you have the receiver components you expected.

Table 158: Component Types

Binary	ASCII	Description
0	UNKNOWN	Unknown component
1	GPSCARD	OEM6 family component
2	CONTROLLER	Reserved
3	ENCLOSURE	OEM card enclosure
8	USERINFO	Application specific information
12	OEM6FPGA	OEM638 FPGA version
13	GPSCARD2	Second card in a ProPak6
14	BLUETOOTH	Bluetooth component in a ProPak6
15	WIFI	WiFi component in a ProPak6
16	CELLULAR	Cellular component in a ProPak6

Binary	ASCII		Description
4-7	Reserved OmniSTAR CAN[a] interface board		
981073920 (0x3A7A0000)	DB_HEIGHTMODEL		Height/track model data
981073921 (0x3A7A0001)	DB_USERAPP		User application firmware
981073925 (0x3A7A0005)	DB_USERAPPAUTO		Auto-starting user application firmware

    a. Please refer to the Acronyms page on our website at http://www.novatel.com.

Table 159: VERSION Log Field Formats

Field Type	Field Format (ASCII)	Description
hw version	P-RS-CCC	Hardware version: P = hardware platform (for example, OEM628) R = hardware revision (for example, 6.00) S = processor revision (for example, A) [a] CCC = COM port configuration (for example, 22T) [b]
sw version	OEM0603xxRN0000	Software Version: OEM06 = the product 03 = the feature release xx = the maintenance release number
boot version	OEM0603xxRGB000	Boot Version: OEM06 = the product 03 = the feature release (content may not be the same as the software version) xx= the maintenance release number
comp date	YYYY/Mmm/DD	YYYY = year Mmm = month DD = day (1 - 31)
comp time	HH:MM:SS	HH = hour MM = minutes SS = seconds

    a. This field may be empty if the revision is not stamped onto the processor.

    b. One character for each of the COM ports 1, 2, and 3. Characters are: 2 for RS-232, 4 for RS-422, T for LV-TTL, and X for user-selectable (valid for COM1 of certain products). Therefore, the example is for a receiver that uses RS-232 for COM 1 and COM 2 and LV-TTL for COM 3.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	VERSION header	Log header		H	0
2	# comp	Number of components (cards, and so on)	Long	4	H
3	type	Component type (see *Table 158, Component Types* on *page 742*)	Enum	4	H+4

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
4	model	OEM6 firmware model number e.g., G1SBOGTTO indicates the receiver's current model functionality	Char[16]	16	H+8
5	psn	Product serial number	Char[16]	16	H+24
6	hw version	Hardware version, see *Table 159, VERSION Log Field Formats* on *page 743*	Char[16]	16	H+40
7	sw version	Firmware software version, see *Table 159*	Char[16]	16	H+56
8	boot version	Boot code version, see *Table 159*	Char[16]	16	H+72
9	comp date	Firmware compile date, see *Table 159*	Char[12]	12	H+88
10	comp time	Firmware compile time, see *Table 159*	Char[12]	12	H+100
11...	Next component offset = H + 4 + (#comp x 108)				
12	xxxx	32-bit CRC (ASCII and Binary only)	Ulong	4	H+4+ (#comp x 108)
13	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

3.2.178 *WIFIAPSTATUS*
Wi-Fi AP status

OEM Platform: ProPak6

This log displays the status of the Wi-Fi module when running as an Access Point (AP).

When the `WIFICONFIG` command (see *page 356*) is used to set the operational MODE to AP, the Wi-Fi controller powers on and the radio is automatically enabled. When the WIFICONFIG command is used to set the STATE to DISABLED or OFF, the Wi-Fi radio is shut off and then the entire controller is shut off correspondingly.

This is a typical transition of states (listed in *Table 160, Wi-Fi AP States* on *page 746*) during normal operation when the Wi-Fi module is powered off then turned on as an active AP. Turning on: OFF → POWERUP → DISABLED → ENABLING → ENABLED

Message ID: 1666
Log Type: Asynchronous

Recommended Input:

> log wifiapstatusa onchanged

ASCII Examples:

The following is an example of state transitions when AP mode is selected, powered on, enabled with no clients connected, and finally connected to by one client (STA) with its MAC address.

```
#WIFIAPSTATUSA,COM1,0,85.5,UNKNOWN,0,4.454,00000020,C981,12312;1,
OFF,"",0*4DF8DCF1

#WIFIAPSTATUSA,COM1,0,84.5,FINESTEERING,1750,162936.127,00000020,
C981,12312;1,POWERUP,"",0*30D2EC08

#WIFIAPSTATUSA,COM1,0,66.5,FINESTEERING,1750,162948.116,00000020,
C981,12312;1,DISABLED,"",0*2C8AF76C

#WIFIAPSTATUSA,COM1,0,85.0,FINESTEERING,1750,162948.120,00000020,
C981,12312;1,ENABLING,"",0*F2D8C440

#WIFIAPSTATUSA,COM1,0,85.0,FINESTEERING,1750,162948.127,00000020,
C981,12312;1,ENABLED,"02:21:66:00:42:56",0*E2BDB713

#WIFIAPSTATUSA,COM1,0,82.0,FINESTEERING,1750,163029.266,00000020,
C981,12312;1,ENABLED,"02:21:66:00:42:56",1,"E8:92:A4:F1:D4:DC",65.0
*ADC1B673
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	WIFIAPSTATUS header	Log header		H	0
2	State	Wi-Fi Access Point State. See *Table 160, Wi-Fi AP States* on *page 746*	Enum	4	H
3	BSSID	Basic Service Set ID used for this AP	String [Max 20]	Variable[a]	H+4
4	AP ID	ID of the active Access Point Profile, defaults to WIFIAPD_1 if omitted.	Enum	4	Variable Max: H+24
5	#Stations	Number of clients connected to the AP. Indicates the number of records to follow.	Ulong	4	Variable Max: H+28
6	STA MAC Address	802.11 Mac address of the client (STA).	String [Max 20]	Variable[a]	Variable Max: H+32
7	STA Link Rate	Negotiated link rate for the client (STA), MBit/sec	Float	4	Variable Max: H+52
8...	Next station offset H+Max[32]+(#stations*Max[24])				
9	xxxx	32-bit CRC (ASCII and Binary only)	-	-	-
10	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 160: Wi-Fi AP States

Binary	ASCII	Description
1	DISABLED	Hardware/Software initialized; RF inactive (radio is off). Not yet acting as a Wi-Fi Access Point
2	ENABLING	Activating RF, enabling Access Point
3	ENABLED	RF active, Access Point is sending beacons, servicing STAs
4	DISABLING	Returning to DISABLED state
5	ERROR	Wi-Fi module is in an error state
6	OFF	Hardware Off
7	POWERUP	Powering up hardware, initializing Wi-Fi Stack
8	POWERDOWN	Uninitializing Wi-Fi stack, powering down hardware

3.2.179 WIFICLISCANRESULTS
Wi-Fi AP scan results

OEM Platform: ProPak6

After Wi-Fi has been enabled in Client mode (see the WIFICONFIG command on *page 356*) and scanning has been initiated (see the WIFICLICONTROL command on *page 354*), this log displays the results of the scan function. Wi-Fi access points that were detected during the scan are shown.

Message ID: 1616
Log Type: Asynchronous

Recommended Input:

> log wificliscanresultsa onnew

ASCII Example:

```
#WIFICLISCANRESULTSA,COM1,0,82.5,FINESTEERING,1745,449323.795,
00000020,DE56,45001;4,INFRASTRUCTURE,"NOVA-CORP",
"78:19:F7:68:9A:BC","802.11X","AES_CCMP","2.4
GHZ",6,3AFF,00000000,00000000,0000FFFF,-74,INFRASTRUCTURE,
"NOVA-GUEST","78:19:F7:68:9A:BD","OPEN","NONE","2.4 GHZ",6,3AFF,
00000000,00000000,0000FFFF,-73,INFRASTRUCTURE,"NOVA-BYOD",
"78:19:F7:68:9A:BE","OPEN","NONE","2.4 GHZ",6,-73,INFRASTRUCTURE,
"PP6TEST","78:19:F7:78:13:06","WPA2_PSK","AES_CCMP","2.4
GHZ",6,3AFF,00000000,00000000,0000FFFF,-73*8CAF7552
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	WIFICLISCAN RESULTS header	Log header		H	0
2	#APs	Number of access points discovered	Ulong	4	H
3	Wi-Fi BSS Type	Wi-Fi BSS Type (see *Table 161, Wi-Fi BSS Types* on *page 748*)	Enum	4	H+4
4	SSID	Service Set Identifier	String [Max 36]	Variable[a]	Variable Max: H+8
5	BSSID	Basic Service Set ID	String [Max 20]	Variable[a]	Variable Max: H+44
6	Authentication type	Open, WEP, WPA2, etc	String [Max 32]	Variable[a]	Variable Max: H+64
7	Encryption protocol	TKIP, AES_CCMP	String [Max 32]	Variable[a]	Variable Max: H+96
8	Band	2.4 or 5 GHz The Wi-Fi radio is limited to 2.4 GHz operation	String [Max 16]	Variable[a]	Variable Max: H+128
9	Channel	Channel	Uint	4	Variable Max: H+144

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
10	Non-HT Rates	List of supported non-High Throughput rates (see *Table 162, Non-HT Rates* on *page 748*)	Ushort	4[a]	Variable Max: H+148
11	HT Rates	List of supported High Throughput rates The HT rates are more complex because they depend on coding mechanism and channel bandwidth, among other factors. The list of supported rates is a bit mask like the previous list of non-HT rates and refers to the MCS (Modulation and Coding Scheme) index. Refer to online documentation (http://mcsindex.com/) for the MCS index meanings.	Ulong[3]	12	Variable Max: H+152
12	RSSI	Received Signal Strength Indicator	Int	4	Variable Max: H+164
13...	Next AP offset H+4+(#APs*Max[64])				
14	xxxx	32-bit CRC (ASCII and Binary only)	-	-	-
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 161: Wi-Fi BSS Types

Wi-Fi BSS Type		Description
Binary	**ASCII**	
0	UNKNOWN	Unknown network
1	INFRASTRUCTURE	Infrastructure network
2	INDEPENDENT	Ad-hoc network

Table 162: Non-HT Rates

Non-HT Rates (Mask)	Rate
0x0001	1 Mbps
0x0002	2 Mbps
0x0004	5.5 Mbps
0x0008	6 Mbps
0x0010	9 Mbps
0x0020	11 Mbps
0x0040	12 Mbps
0x0080	18 Mbps

Non-HT Rates (Mask)	Rate
0x0100	22 Mbps
0x0200	24 Mbps
0x0400	33 Mbps
0x0800	36 Mbps
0x1000	48 Mbps
0x2000	54 Mbps

3.2.180 WIFICLISTATUS
Wi-Fi client connection status

OEM Platform: ProPak6

This log displays the status of the Wi-Fi module when running as a client.

When the WIFICONFIG command (see *page 356*) is used to set the operational MODE to Client, the Wi-Fi controller powers on but the radio remains off until the WIFICONFIG command is used to set the STATE to ENABLED. When the WIFICONFIG command is used to set the STATE to DISABLED or OFF, the Wi-Fi radio is shut off and then the entire controller is shut off.

This is a typical transition of states (listed in *Table 163, Wi-Fi Client State* on *page 752*) during normal operation when the Wi-Fi module is powered off, turned on as a client, then connected to an AP. Note that if some transitional states occur internally very rapidly, they might not be shown in output logs.
Turning on: OFF → POWERUP → DISABLED → ENABLING → DISCONNECTED → ASSOCIATING → IPADDRESS → CONNECTED

Message ID: 1613
Log Type: Asynchronous

Recommended input:

> log wificlistatusa once

ASCII Example:

```
#WIFICLISTATUSA,COM1,0,85.5,UNKNOWN,0,708.095,004C4020,71A1,45094;
OFF,"02:21:66:00:42:56",FALSE,-1.0,65.0,NONE,"","","",0,-99,NONE,0
*8892694C
#WIFICLISTATUSA,COM1,0,84.5,UNKNOWN,0,738.736,004C4020,71A1,45094;
POWERUP,"02:21:66:00:42:56",FALSE,-1.0,65.0,NONE,"","","",0,-99,
NONE,0*CDB320A1
#WIFICLISTATUSA,COM1,0,67.0,UNKNOWN,0,747.011,004C4020,71A1,45094;
DISABLED,"02:21:66:00:42:56",FALSE,-1.0,65.0,NONE,"","","",0,-99,
NONE,0*16C6C316
#WIFICLISTATUSA,COM1,0,84.0,UNKNOWN,0,747.017,004C4020,71A1,45094;
DISCONNECTED,"02:21:66:00:42:56",FALSE,-1.0,65.0,NONE,"","","",0,
-99,AUTH_UNSUPPORTED,1064*6468065B
#WIFICLISTATUSA,COM1,0,81.5,UNKNOWN,0,2648.272,004C4020,71A1,45094;
ASSOCIATING,"02:21:66:00:42:56",FALSE,-1.0,65.0,1,"NETWORK23","",
"",0,-99,NONE,0*CAB8D09C
#WIFICLISTATUSA,COM1,0,37.5,UNKNOWN,0,2649.516,004C4020,71A1,45094;
CONNECTED,"02:21:66:00:42:56",FALSE,54.0,65.0,1,"NETWORK23",
"C0:3F:0E:8A:C6:26","",7,-99,NONE,0*C34C0201
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	WIFICLISTATUS header	Log header		H	0
2	State	Wi-Fi Client State (see *Table 163, Wi-Fi Client State* on *page 752*)	Enum	4	H
3	MAC address	MAC address of the client interface	String [Max 20]	Variable[a]	Variable Max: H+ 4
4	Scan In Progress	TRUE when a scan for access points is running (TRUE=1)	Bool	4	Variable Max: H+24
5	Link Speed	Current Link Speed	Float	4	Variable Max: H+28
6	Link Speed Max	Negotiated Link Speed	Float	4	Variable Max: H+32
7	Network ID	Wi-Fi Network ID (See *Table 164, Wi-Fi Network ID* on *page 752*)	Enum	4	Variable Max: H+36
8	SSID	Service Set Identifier of the current network, if any	String [Max 36]	Variable[a]	Variable Max: H+40
9	BSSID	Basic Service Set ID of the current network	String [Max 20]	Variable[a]	Variable Max: H+76
10	Frequency Band	Frequency band used	String [Max 16}	Variable[a]	Variable Max: H+96
11	Channel	Channel used	Uint	4	Variable Max: H+112
12	RSSI	Receiver Signal Strength Indicator	Int	4	Variable Max: H+116
13	Error	Error associated with the last failed operation (see *Table 165, Wi-Fi Client Error* on *page 753*	Enum	4	Variable Max: H+120
14	Reserved		Ulong	4	Variable Max: H+124
15	xxxx	32-bit CRC (ASCII and Binary only)	-	-	-
16	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

a. In the binary case, each string field needs to be NULL terminated and additional bytes of padding added to maintain 4-byte alignment, up to the maximum defined by the string size. The next defined field starts immediately at the next 4-byte alignment following the NULL.

Table 163: Wi-Fi Client State

Value	Name	Description
0	Unknown	Internal state
1	Disabled	Hardware/Software initialized; RF inactive (radio is off). Not yet acting as a Wi-Fi client
2	Enabling	Activating RF; enabling client
3	Disconnected	RF enabled; not connected to AP
4	Associating	Associating / Authenticating to an AP
5	IPAddress	Obtaining IP address using DHCP, or applying static IP configuration
6	Connected	Fully configured (802.11 + IP); interface is ready for use
7	Disconnecting	Disconnecting from AP
8	Disabling	Disabling RF, hardware returning to initialized state
9	Error	Hardware non-responsive; client will be reset shortly.
10	Calibration	
11	Off	Hardware is fully powered off (radio and controller)
12	Powerup	Hardware controller is turning on (radio is still off)
13	Powerdown	Hardware controller is turning off (radio will be shut off)

Table 164: Wi-Fi Network ID

Value	Network	Description
0	NONE	
1	1	Network 1
2	2	Network 2
3	3	Network 3
4	4	Network 4
5	5	Network 5
6	6	Network 6
7	7	Network 7
8	8	Network 8
9	9	Network 9
10	10	Network 10
11	11	Network 11
12	12	Network 12
13	13	Network 13

Value	Network	Description
14	14	Network 14
15	15	Network 15
16	16	Network 16
17	GLOBAL	Global setting, applicable to all networks

Table 165: Wi-Fi Client Error

Value	Error	Description
0	NONE	No error
1	GENERAL	General/unknown error
2	HARDWARE	Hardware failure
3	INTERNAL	Internal software error
4	BUSY	Software is busy
5	BSS_UNAVAILABLE	The access point (BSS) is offline/not available/not responding
6	ASSOC_DENIED	AP denied association request
7	AUTH_FAILURE	Authentication failed, possibly due to incorrect passphrase / key
8	AUTH_UNSUPPORTED	Authentication method not supported: Network configuration is not correct. e.g., open authentication configured for WPA2 AP
8	DISASSOCIATION	AP has forced disassociation
10	TIMEOUT	Timeout during operation

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 166: 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

Table 166: Response Messages

CHANNEL IS INVALID	20	The selected channel is invalid
REQUESTED RATE IS INVALID	21	The requested rate is invalid
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