Hemisphere GNSS Technical Reference Guide v1.07.1

Current Version: v1.07.1 April 27, 2017

Table of Contents

Contents

Introduction

The purpose of the GNSS Technical Reference is to serve as a resource for software engineers and system integrators engaged in the configuration of GNSS receivers. It may also be of use to persons with knowledge of the installation and operation of GNSS navigation systems.

This reference covers features, commands, logs, and operating modes for a variety of Hemisphere GNSS products: not all aspects described apply to all products.

Information is provided as follows:

- Quick Start provides basic information to get you started using your Hemisphere GNSS receiver
- GNSS Technology and Platforms provides information on the GNSS engine, GNSS solutions, and GNSS platforms
- Receiver Operation introduces general operational features of the receiver, receiver operation modes, and default operation parameters
- Commands and Messages are grouped by their type (General, GNSS, e-Dif, Data, RAIM, etc.) and for each type the commands or messages are initially listed in a table with a brief description. Each command and message is then described in detail in separate topics.
- Resources provides resources for additional information
- Change History provides a list of all topics updated in a release and a short description of each change
- Troubleshooting provides troubleshooting advice

Topic Last Updated: v1.07 / February 16, 2017

Quick Start

This topic provides basic information to get you started using your Hemisphere GNSS receiver.

- What is my receiver type?
  Send the JT command.

- How do I load firmware onto my receiver and why would I do this?
  Use RightARM. Loading firmware allows you to run application specific capabilities.

- What is my current receiver configuration?
  Send the JSHOW query.
  For Vector products send the JATT,SUMMARY query.

- What commands are supported by my receiver?
  Find out what GNSS engine is in your receiver (issue JT command) then go to the Overview topic for commands supported by that GNSS engine.

- How do I send a command to my receiver?
  Connect receiver to a PC and use a terminal program (such as HyperTerminal) or Hemisphere GNSS' PocketMax or SLXMon. For more information refer to the User Guide for your product.

- How do I turn on data messages (such as GPGGA) for a receiver?
  See Configuring the Data Message Output.

Topic Last Updated: v1.07 / February 16, 2017

GNSS Technology and Platforms

GNSS Engine

GNSS Engine Overview

The GNSS engine is always operating regardless of the DGNSS mode of operation. The following sections describe the general operation of the receiver.

- Satellite Tracking
- Positioning Accuracy
- Update Rates

Both the GNSS and SBAS operation of the receiver module features automatic operational algorithms. When powered for the first time, the receiver system performs a "cold start," which involves acquiring the available GNSS satellites in view and the SBAS differential service. To do this, the receiver needs a compatible GNSS antenna connected that offers a relatively clear, unobstructed view of the sky. While you can often achieve this indoors with an antenna placed against a window, you may need to place the antenna outside, for example on a roof or a short distance away from the building.

If SBAS is not available in a particular area, an external source of RTCM SC-104 differential correction may be used. If an external source of correction data is needed, the external source needs to support an eight data bit, no parity and one stop bit configuration (8-N-1). See also SBAS Overview.

Topic Last Updated: v1.07 / February 16, 2017

Satellite Tracking

The receiver automatically searches for GNSS satellites, acquires the signal, and manages the associated navigation information required for positioning and tracking. This is a hands-free mode of operation. Satellite acquisition quality is described as a signal-to-noise ratio (SNR) and the higher the SNR, the better the signal reception quality. SNR information is provided by the receiver through the use of NMEA 0183 data messages available via its multiple serial ports.

Topic Last Updated: v1.07 / February 16, 2017

Positioning Accuracy

The receiver is a sub-meter product with 95% horizontal accuracy under ideal conditions.

To determine the positioning performance of the receiver, Hemisphere GNSS gathers a 24-hour data set of positions in order to log the diurnal environmental effects and full GPS constellation changes. Data sets shorter than 24 hours tend to provide more optimistic results.

The horizontal performance specification of 95% accuracy is, as stated above, based on ideal conditions. In reality, obstruction of satellites, multipath signals from reflective objects, and operating with poor corrections will detract from the receiver's ability to provide accurate and reliable positions. Differential performance can also be compromised if the receiver module is used in a region without sufficient ionospheric coverage.
Further, if external corrections are used, the baseline separation between the remote base station antennas can affect performance.

Since the receiver will be used in the real world, blockage of the line of sight to SBAS satellites is often inevitable. The COAST function provides solace from obstruction of any differential correction source (SBAS, Beacon, RTCM, Atlas, RTK, e-Dif) for 30 to 40 minutes depending on the amount of tolerable performance drift. In fact, our receivers will COAST when differential correction is lost no matter what the differential source is: SBAS, Beacon, RTCM, Atlas, RTK, or e-Dif.

The estimated positioning precision is accessible through the use of NMEA 0183 command responses as described Commands and Messages.

Because the receiver cannot determine accuracy with respect to a known location in real time (so is traditionally performed in post-mission analyses), the precision numbers are relative in nature and are only approximates.

Topic Last Updated: v1.06 / March 10, 2015

Update Rates

The update rate of each NMEA 0183 and binary message of the receiver can be set independently with a maximum that is dependant upon the message type. For example, some messages have a 1 Hz maximum while other messages have a 20 Hz maximum. The higher update rates, such as 20 Hz, are an option and can be obtained at an additional cost.

Higher update rates are valuable for applications where:

- Higher speeds are present such as in aviation
- You have manual navigational tasks such as in agricultural guidance
- You have an automated or autonomous navigational task such as in robotics or machine control

Keep the following in mind regarding message rates:

- Some messages can only be OFF or ON (0 or 1Hz)
  Example: $JASC,RTCM3,1
- Some messages can only be 0 or 1 Hz, but will come out once first, then only if they change
  Example: $JASC,BIN95,1
- Messages that are available at other rates can be set to rates SLOWER than 1 Hz *(see Note 1 below*)
  Example: $JASC,GPGGA,0.1
- If the receiver is subscribed to 10 or 20Hz, the receiver can log at rates FASTER than 1 Hz *(see Note 2 below)*
  Example: $JASC,GPGGA,5

Note 1: Slower than 1 Hz.

Use the following guidelines:

To log once every_____ seconds	Use JASC,xxxx,______
2	0.5
3	0.3333
4	0.25
5	.2
6	0.1667
7	0.1429
8	0.125
9	0.1111
10	0.1
15	0.0667
20	0.05
25	0.04
40	0.025
50	0.02
100	0.01
120	0.0083

Rates not listed above may be possible but may not log on integer seconds. Users should test to see if the results are acceptable for their application.

Note 2: Faster than 1Hz, if subscribed.

Acceptable rates are 1, 2, 4, 5, 10 or 20 Hz. Using rates other than those listed will result in data appearing in a rate similar to the rate requested, but the data times will be quantized to 0.05 second resolution. This is due to the receiver's internal computing rate of 20 Hz. Time resolution is 0.05 seconds even if the receiver is only subscribed for 10 Hz data. Quantizing may result in a slightly different number of messages per minute than expected. For example, 3 Hz data produces approximately 172 messages per minute due to quantizing, instead of the expected 180 messages.

Using rates other than a factor of 20 Hz may result in quantized data. Regardless, the data in the message is referenced to the time of the message. For example, 3 Hz data may appear at a time of 0.30 seconds; the data is referenced to 0.3 seconds, not 0.333333 seconds.

Topic Last Updated: v1.00 / August 11, 2010

DGNSS Solutions

COAST Technology

Crescent and Eclipse OEM boards feature Hemisphere GNSS' exclusive COAST technology that enables Hemisphere GNSS Crescent and Eclipse receivers to utilize old DGPS correction data for 40 minutes or more without significantly affecting positioning quality.

Note: Crescent refers to Crescent, Crescent Vector

When using COAST, these receivers are less likely to be affected by differential signal outages due to signal blockages, weak signals, or interference.

Note: To obtain a full set of SBAS corrections, the COAST technology provides the following benefits:

- Accurate and minimal position drift during temporary loss of differential signal corrections
- Sub-meter accuracy up to 40 minutes after differential signal loss
- Outstanding performance in environments where maintaining a consistent differential link is difficult
- It is standard with Crescent and Eclipse GPS receiver technology

Topic Last Updated: v1.06 / March 10, 2015

SBAS

SBAS Overview

The following topics describe the general operation and performance monitoring of the Space-Based Augmentation System (SBAS) demodulator within the receiver module:

- Automatic tracking
- Performance
- WAAS
- WAAS DGPS
- WAAS Signal Information
- WAAS Reception
- WAAS Coverage

Topic Last Updated: v1.00 / August 11, 2010

SBAS Automatic Tracking

The SBAS demodulator featured within the receiver automatically scans and tracks multiple SBAS satellite signals, as specified by the JWAASPRN command (defaulted to WAAS PRN 135 and 138, suitable for use in North America).

If the default satellites become disabled, the receiver automatically tracks different satellites. This automatic tracking enables you to focus on other aspects of your application rather than ensuring the receiver is tracking SBAS correctly.

The SBAS demodulator features two-channel tracking that enhances the ability to maintain acquisition on an SBAS signal satellite in regions where more than one satellite is in view.

This redundant tracking approach results in more consistent signal acquisition in areas where signal blockage of either satellite is possible.

SBAS Performance

SBAS performance is described in terms of bit error rate (BER). The SBAS receiver requires a line of sight to the SBAS satellite to acquire a signal.

The BER number indicates the number of unsuccessfully decoded symbols in a moving window of 2048 symbols. Due to the use of forward error correction algorithms, one symbol is composed of two bits. The BER value for both SBAS receiver channels is available in the RD1 message.

A lower BER indicates data is being successfully decoded with fewer errors, providing more consistent throughput. The BER has a default no-lock of 500 or more. As the receiver begins to successfully acquire a signal, a lower BER results. For best operation, this value should be less than 150 and ideally less than 20.

SBAS broadcasts an ionospheric map on a periodic basis and it can take up to five minutes to receive the map on startup. Until it downloads the SBAS map the receiver uses the broadcast ionosphere model, which can result in a lower performance compared to when the map has been downloaded. This is the case for any GNSS product supporting SBAS services.

WARNING: **When the map has been downloaded, you may observe a position jump due to the potentialdifference between the GPS ionospheric model and the ionosphere SBAS map. To minimize the impact of this issue on the use of the receive wait up to five minutes before using the receiver or issue the JQUERY,GUIDE command to 'ask' the receiver if it feels the performance will be sufficient for operation.**

Topic Last Updated: v11.07 / February 16, 2017

WAAS

The US Federal Aviation Administration developed the Wide Area Augmentation System (WAAS) to provide accurate positioning to the aviation industry. In addition to providing a high quality and accurate service for this industry, the service is available free of charge to civilians and markets in North America.

Other government agencies have developed similar WAAS-compatible systems for their respective geographic regions.

- Europe - the European Space Agency, the European Commission and EUROCONTROL jointly developed the European Geostationary Navigation Overlay Service (EGNOS)
- Japan - the MTSAT Satellite-based Augmentation System (MSAS) was developed by the Japan Civil Aviation Bureau (JCAB)
- India - the Airport Authority of India and the Indian Space Research Organization (ISRO) are deploying the GPS Aided Geo Augmented Navigation system (GAGAN)

These compatible augmentation systems fall into a broader category often referred to as Space Based Augmentation System (SBAS). The receiver is capable of receiving correction data from all WAAS- compatible SBAS**.**

Topic Last Updated: v1.00 / August 11, 2010

WAAS DGPS

WAAS differential, and other compatible SBAS, use a state-based approach in their software architecture. These services take in reference data from a network of base stations and endeavor to model the sources of error directly, rather than computing the sum impact of errors upon observed ranges. The advantage of this approach is that the error source can be more specifically accounted for during the correction process.

Specifically, WAAS calculates separate errors for the following:

- Ionospheric error
- GPS satellite timing errors
- GPS satellite orbit errors

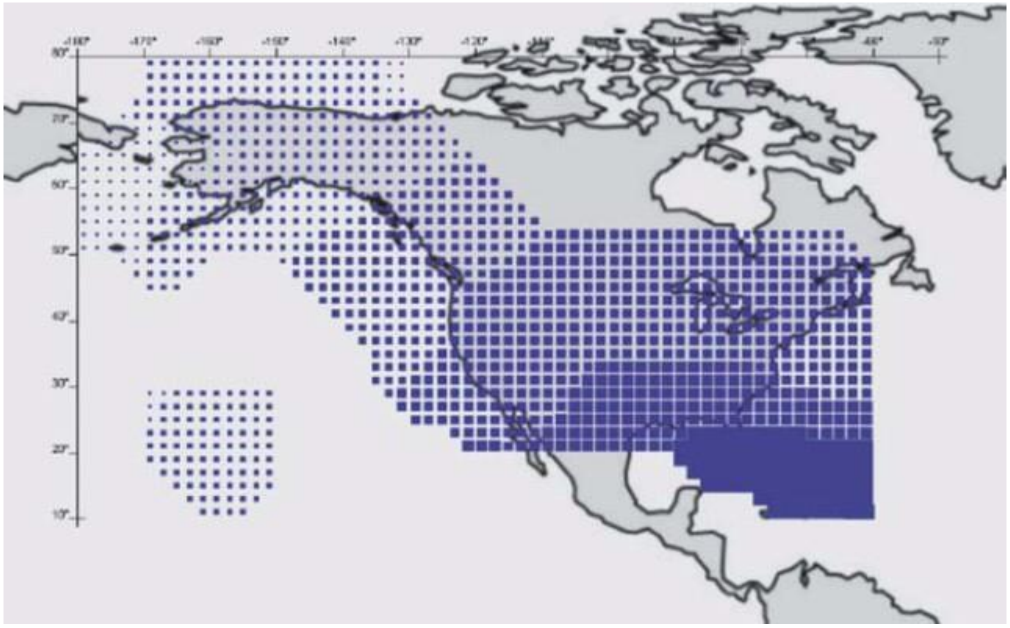
Provided that a GNSS satellite is available to the WAAS reference station network for tracking purposes, orbit and timing error corrections will be available for that satellite. Ionospheric corrections for that satellite are only available if the signal passes through the ionospheric map provided by WAAS, which covers most of North America.

To improve the ionospheric map provided by WAAS, the receiver extrapolates information from the broadcast ionospheric coverage map, extending its effective coverage. This allows the receiver to be used successfully in regions that competitive products may not. This is especially important in Canada for regions north of approximately 54° N latitude and for outer regions of the Caribbean.

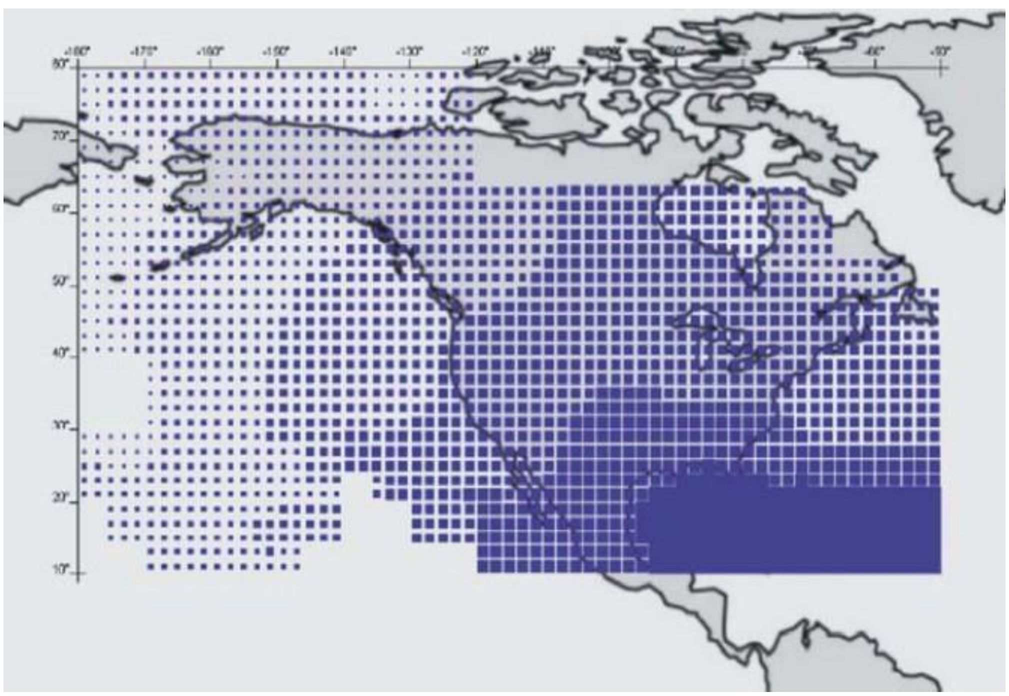
The process of estimating ionospheric corrections beyond the WAAS broadcast map is not as good as having an extended WAAS map and accuracy degradation may occur.

The map links below depict the broadcast WAAS ionospheric map coverage and the Hemisphere GNSS extrapolated version, respectively. As the two maps show, the Hemisphere GNSS extrapolated version’s coverage is greater in all directions, enhancing usable coverage.

- Broadcast WAAS ionospheric correction map



- Extrapolated WAAS ionospheric correction map



Topic Last Updated: v1.07 / February 16, 2017

WAAS Signal Information

WAAS and other SBAS systems transmit correction data on the same frequency as GPS, allowing the use of the same receiver equipment used for GPS. Another advantage of having WAAS transmit on the same frequency as GPS is that only one antenna element is required.

Topic Last Updated: v1.00 / August 11, 2010

WAAS Reception

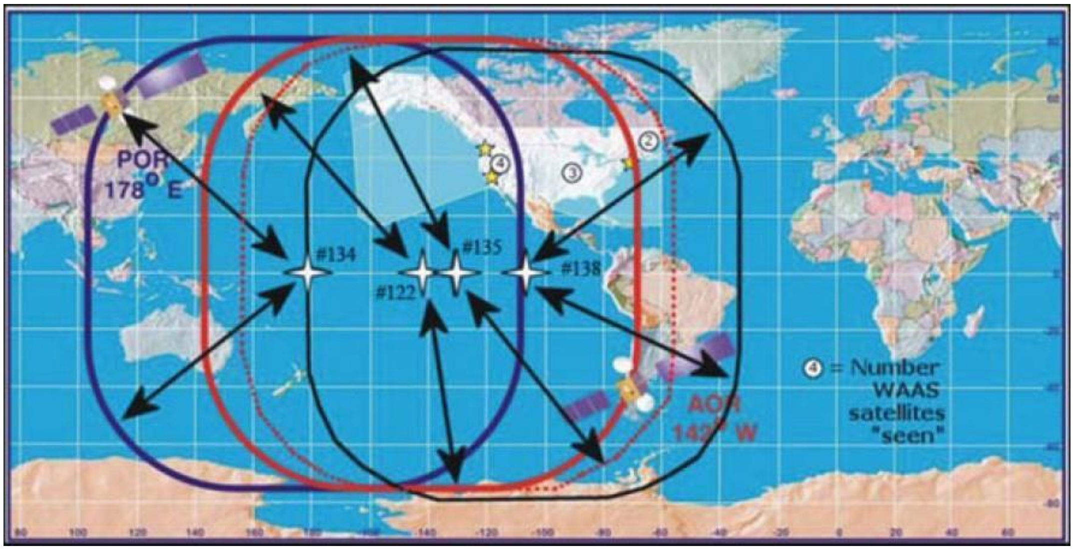
Since WAAS broadcasts on the same frequency as GPS, the signal requires a line of site in the same manner as GPS to maintain signal acquisition.

Because of their locations, SBAS satellites may appear lower on the horizon than GPS satellites—it depends on the geographic position on land. When using WAAS correction data, the receiver can provide the azimuth and elevation of all satellites to aid in determining their position with respect to the antenna.

Topic Last Updated: v1.00 / August 11, 2010

WAAS Coverage

The figure below depicts the current WAAS coverage provided by the geostationary satellites.



The WAAS satellites are identified by their pseudorange number (PRN). In some areas, two or more satellites may be visible.

Note: Signal coverage may be present in some areas without either sufficient ionospheric map coverage or satellites with valid orbit and clock corrections. In such cases performance may be degraded compared to areas fully covered by the WAAS ionospheric coverage.

Topic Last Updated: v1.00 / August 11, 2010

EGNOS

The European Geostationary Navigation Overlay Service (EGNOS) uses multiple geostationary satellites and a network of ground stations to transmit differential correction data for public use. EGNOS is currently located over the Atlantic Ocean and Africa.

Because of their location over the equator, these satellites may appear lower over the horizon as compared to GPS satellites - it depends on the geographic position on the land. In regions where the satellites appear lower on the horizon, they may be more susceptible to being masked by terrain, foliage, buildings or other objects, resulting in signal loss. Increased distance from the equator and the satellite's longitude cause the satellite to appear lower on the horizon. Hemisphere GNSS's COAST technology helps alleviate this problem by maintaining system performance when EGNOS signal loss occurs for extended periods of time. More information on COAST technology is provided later in this chapter.

The figure below shows approximate EGNOS coverage provided by the satellites. Virtually all of Europe, part of Northern Africa, and part of the Middle East is covered with at least one signal. Most of Europe is covered by three signals.

Note: Increased distance from the equator and the satellite's longitude cause the satellite to appear lower on the horizon. Although a good amount of signal coverage is shown in northern latitudes for EGNOS, it may not be usable because of its low elevation angle and the potential for it to be obstructed. Testing of the system in the area of its use is recommended to ensure that the signal is sufficiently available.

Topic Last Updated: v1.06 / March 10, 2015

MSAS

The MTSAT Satellite-based Augmentation System (MSAS) is currently run by the Japan Meteorological Agency (JMA). MSAS provides GPS augmentation information to aircraft through MTSAT (Multi-functional Transport Satellite) located approximately 36000 km above the equator (geostationary earth orbit).

MSAS generates GPS augmentation information by analyzing signals from GPS satellites received by monitor stations on the ground. This augmentation information consists of GPS-like ranging signal and correction information on GPS errors caused by the satellites themselves or by the ionosphere.

The MSAS signal provides accurate, stable, and reliable GPS position solutions to aircraft, resulting in a considerable improvement in the safety and reliability of GPS positioning. This enables aviation users who are under very strict safety regulations to use GPS positioning as a primary navigation system.

Visit http://www.jma.go.jp/jma/jma-eng/satellite/ for more information on MSAS and MTSAT.

Topic Last Updated: v1.00 / August 11, 2010

GAGAN

The GPS Aided Geo Augmented Navigation system (GAGAN) is currently under deployment by the Indian government and is anticipated to be operational by 2011. It operates similarly to the other SBAS regions described previously and will broadcast on one geostationary satellite (PRN 127) over the Western portion of the Indian Ocean. GAGAN should be visible in India at elevation angles in excess of 50º above the horizon. This will provide an excellent correction source in virtually all areas of the subcontinent.

Topic Last Updated: v1.00 / August 11, 2010

Radiobeacon

Radiobeacon Overview

Many marine authorities, such as Coast Guards, have installed networks of radiobeacons that broadcast DGPS corrections to their users. With increasing use of these networks for terrestrial applications, there is increasing densification of these networks inland.

Topic Last Updated: v1.00 / August 11, 2010

Radiobeacon Range

The broadcasting range of a 300 kHz beacon depends on a number of factors, including:

- Transmission power
- Free space loss
- Ionospheric state
- Surface conductivity
- Ambient noise
- Atmospheric losses

Signal strength decreases with distance from the transmitting station, mostly due to spreading loss. This loss is a result of the signal's power being distributed over an increasing surface area as the signal radiates away from the transmitting antenna.

The expected broadcast range also depends on the conductivity of the surface over which it travels. A signal will propagate further over a surface area with high conductivity than over a surface with low conductivity.
Lower conductivity surfaces, such as dry, infertile soil, absorb the power of the transmission more than higher conductivity surfaces, such as sea water or arable land.

A radiobeacon transmission has three components:

1. Direct line-of-sight wave
   The line-of-sight wave is insignificant beyond visual range of the transmitting tower and does not have a substantial impact upon signal reception.

2. Ground wave
   The ground wave portion of the signal propagates along the surface of the earth, losing strength due to spreading loss, atmospheric refraction and diffraction, and attenuation by the surface over which it travels (dependent upon conductivity).

3. Sky wave
   Depending on its reflectance, this skyward portion of the beacon signal may bounce off the ionosphere and back to Earth, causing reception of the ground wave to fade. Fading—which may cause reception to fade in and out—occurs when the ground and sky waves interfere with each other. This problem usually occurs in the evening when the ionosphere becomes more reflective and usually on the edge of coverage areas. Fading is not usually an issue with overlapping coverage areas of beacons and their large overall range.

Atmospheric attenuation plays a minor part in signal transmission range because it absorbs and scatters the signal. This type of loss is the least significant of those described.

Topic Last Updated: v1.00 / August 11, 2010

Radiobeacon Reception

Various noise sources affect beacon reception and include:

- Engine noise
- Alternator noise
- Noise from power lines
- DC to AC inverting equipment
- Electric devices such as CRTs, electric motors, and solenoids

Noise generated by these types of equipment can mask the beacon signal, reducing or impairing reception.

Topic Last Updated: v1.00 / August 11, 2010

Radiobeacon Antenna Location

When using the internal beacon receiver as the correction source, antenna location will influence the performance of the internal beacon receiver.

A good location will:

- Have a clear view of the sky (important for GNSS, WAAS, and Atlas signal reception)
- Be at least three feet away from all forms of transmitting antennas, communications, and electrical equipment, to reduce the amount of noise present at the antenna
- Be the best for the application, such as the center line of the vehicle or vessel (the position calculated by the beacon receiver is measured to the center of the antenna)
- Not be in areas that exceed specified environmental conditions

Topic Last Updated: v1.07 / February 1, 2017

Radiobeacon Coverage

The figure below shows the approximate radiobeacon coverage throughout the world. Light shaded regions denote current coverage, with beacon stations shown as white circles. The world beacon networks continue to expand. For more current coverage, visit the Hemisphere GNSS web site at www.hemispheregnss.com.

Topic Last Updated: v1.06 / March 10, 2015

Atlas

Atlas Overview

Atlas services provides correction data to subscribers of the system with the use of a geostationary transponder.

The information broadcast by DGNSS services is based on a network of reference stations— placed at geographically strategic locations—that communicate GNSS correction data to control centers. At the control centers the GNSS correction data is decoded, checked, and repackaged into a proprietary format for transmission to a geostationary Atlas communications satellite. The satellite rebroadcasts the correction information back to earth over a large signal footprint where the Hemisphere GNSS Atlas differential satellite receiver demodulates the data.

L-band DGNSS service signal content is not RTCM SC-104, but a proprietary wide-area signal that's geographically independent. With this service, the positioning accuracy does not degrade as a function of distance to a base station because the data content is not composed of a single base station's information; it is composed of an entire network's information. When the Hemisphere GNSS Atlas receiver demodulates the proprietary signal it converts it into a local-area format for input to the GNSS receiver (standard RTCM SC-104, message Type 1).

The Atlas receiver interpolates corrections from the wide-area signal, specific to the location using Atlas service processing algorithms. The resulting RTCM corrections are those that would be calculated if a reference station were set up at the present location. This type of solution ensures a consistent level of accuracy across the entire coverage area. The GNSS receiver provides position information to the Atlas receiver for Atlas service calculations.

Atlas high precision services are also available. Atlas high precision services require a dual frequency receiver such as the Eclipse to function properly and are approximately three to seven times more accurate than standard Atlas service.

Topic Last Updated: v1.07 / February 16, 2017

Atlas Signal Information

The Atlas signal is a line-of-sight UHF signal that is similar to GNSS. For the Atlas differential receiver to acquire the signal, there must be a line of sight between the antenna and the geostationary communications satellite.

Various Atlas communications satellites are used for transmitting the correction data to Atlas users around the world. When the Atlas receiver has acquired an Atlas signal, the elevation and azimuth are available in the menu system to enable troubleshooting line-of sight problems.

Contact your Atlas service provider for further information on this service.

Topic Last Updated: v1.07 / February 16, 2017

Atlas Reception

Atlas services broadcast at a similar frequency to GNSS and as a result is a line-of-sight system; there must be a line of sight between the antenna and the Atlas satellite for reception of the service.

Atlas services use geostationary satellites for communication. The elevation angle to these satellites is dependent upon latitude. For latitudes higher than approximately 55° North or South, the Atlas signal may be blocked more easily by obstructions such as trees, buildings, and terrain.

Topic Last Updated: v1.07/ February 16, 2017

Atlas Automatic Tracking

The Hemisphere GNSS Atlas receiver features an automatic mode that allows it to locate the best spot beam if more than one is available in a particular region. With this function you do not need to adjust the receiver's frequency. The receiver also features a manual tune mode for flexibility.

See the JFREQ command for more information on automatic and manual tuning.

Topic Last Updated: v1.07 / February 16, 2017

Atlas Receiver Performance

Atlas receivers provide both a lock indicator and a BER (bit error rate) to describe the lock status and reception quality. Both these features depend on a line of sight between the antenna and the geostationary communications satellite broadcasting the Atlas correction information.

Atlas capable Hemisphere GNSS antennas are designed with sufficient gain at low elevation angles to perform well at higher latitudes where the signal power is lower and the satellite appears lower on the horizon. The BER number indicates the number of unsuccessfully decoded symbols in a moving window of 2048 symbols. Because of the use of forward error correction algorithms, one symbol is composed of two bits.

The BER has a default, no-lock value of 500. As the receiver begins to successfully acquire the signal a lower BER results. For best operation this value should be less than 150 and ideally less than 20.

Topic Last Updated: v1.07 / February 16, 2017

Crescent Base Station

Crescent Base Station Overview

The Crescent receiver with e-Dif subscription can operate in a DGPS base station mode. NMEA 0183 commands need to be sent to the receiver to enter this mode. These commands may be automatically issued through customized software or through a simple terminal interface running on a PC, PDA, or data logger. DGPS Base Station Commands provides detailed information on the commands supported by the base station application.

Topic Last Updated: v1.00 / August 11, 2010

Crescent Base Station Startup

When the receiver running the e-Dif application first starts up, it requires a few minutes to gather enough satellite tracking information to model the errors for the future. Once commands are sent to put the receiver into base station mode, corrections will be generated and can be sent via the serial port to rover receivers. In some more challenging GNSS environments, the time required to model errors can take up to 10 minutes. The receiver must be stationary during this process and the antenna for the base station must be secured in a stable location.

Topic Last Updated: v1.07 / February 16, 2017

Crescent Base Station Calibration

Base station calibration is the process of modeling the errors at the base station. Calibration can be performed in either a relative or an absolute sense, depending on positioning needs. Relative positioning provides positions that are accurate to one another but there may be some offset from the true geographical position.

Calibrating for relative positioning is easier than for absolute position since you are not restricted to using a point with known coordinates. Calibrating for absolute positioning mode requires placing the GPS antenna at a known reference location. Care should be taken to use a location that has good sky visibility and is relatively free from obstructions.

Topic Last Updated: v1.00 / August 11, 2010

Crescent Base Station Performance

Base station performance depends primarily on the site location for the base station GNSS antenna. An ideal location would have no obstructions above the height of the antenna, offering a full 180º by 360º view of the sky. In reality, obstructions such as trees, vehicles, people, and buildings nearby both block satellite signals and reflect interfering signals called multipath signals. Multipath degrades the accuracy of the satellite measurements and detracts from the receiver's ability to provide accurate and reliable corrections for the rovers.

For a rover to work optimally, a base station should be near by the rover's area of operation. As distance from the base to the rover increases, the modeling process cannot tune the solution to the exact environmental conditions at the rover's location and the rover's accuracy will not be as good. Best performance is attained when the distance from your base to your rover is less than 50 km (30 miles).
Generally, there is little to no advantage to using a base station if it is more than 300 km (180 miles) from the rover.

Topic Last Updated: v1.07 / February 16, 2017

e-Dif

e-Dif - Extended Differential Option for the Crescent Receiver

The Crescent receiver module is designed to work with Hemisphere GNSS' patented Extended Differential (e-Dif) software. e-Dif is an optional mode where the receiver can perform with differential-like accuracy for extended periods of time without the use of a differential service. It models the effects of ionosphere, troposphere, and timing errors for extended periods by computing its own set of pseudo-corrections.

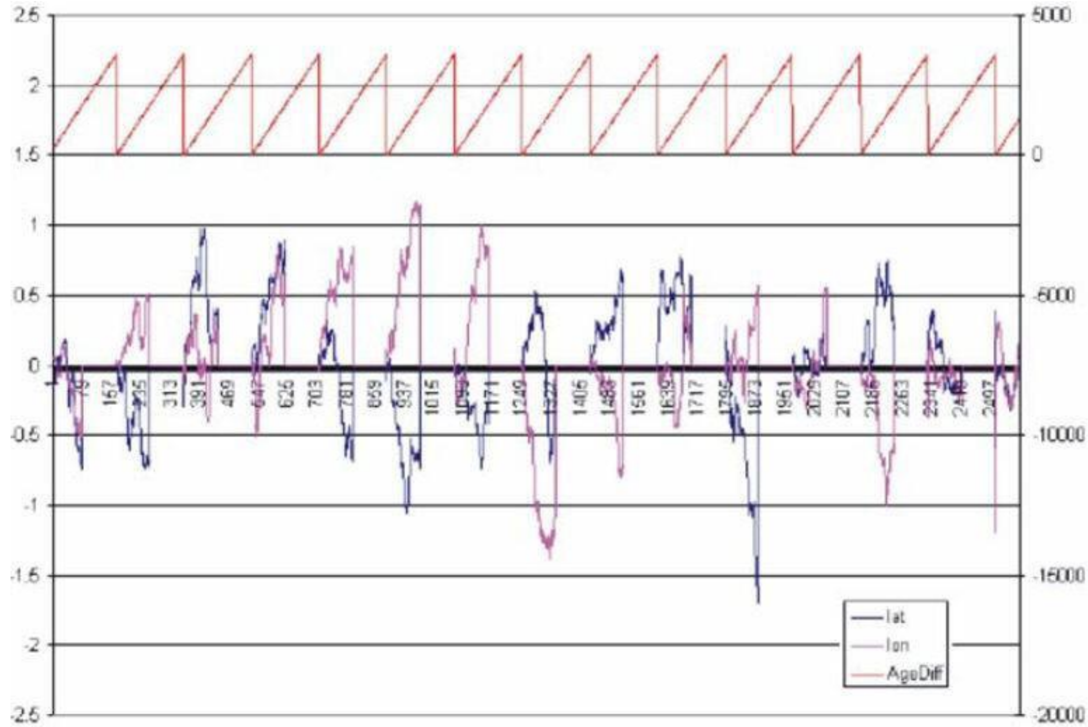
e-Dif may be used anywhere geographically and is especially useful where SBAS networks have not yet been installed, such as South America, Africa, Australia, and Asia. Two things are required to enable e-Dif. First your receiver will require the e-Dif application software to be installed on it. As well, a software key, called a subscription code, is needed for the receiver to use e-Dif. Both can be installed in the field using a PC computer. See Using RightARM to Load Firmware if you need to install the application firmware onto your receiver. To install a subscription code, contact Hemisphere GNSS for a JK command which can be issued to your receiver.

Positioning with e-Dif is jump-free compared to a receiver working with just raw GPS provided the receiver consistently maintains a lock on at least four satellites at one time. The accuracy of positioning will have a slow drift that limits use of the e-Dif for approximately 30 to 40 minutes although it depends on how tolerant the application is to drift as e-Dif can be used for longer periods.

This mode of operation should be tested to determine if it is suitable for the application and for how long the user is comfortable with its use. As accuracy will slowly drift, the point at which to recalibrate e-Dif to maintain a certain level of accuracy must be determined.

The figure below displays the static positioning error of e-Dif while it is allowed to age for fourteen consecutive cycles of 30 minutes. The top line indicates the age of the differential corrections. The receiver computes a new set of corrections using e-Dif during the calibration at the beginning of each hour and modifies these corrections according to its models. After the initialization, the age correspondingly increases from zero until the next calibration.

The position excursion from the true position (the lines centered on the zero axis are northing [dark line] and easting [light line]) with increasing correction age is smooth from position to position; however, there is a slow drift to the position. The amount of drift depends on the rate of change of the environmental errors relative to the models used inside the e-Dif software engine.



Note: You decide how long e-Dif is to function before between calibrations and you should test this operation mode to determine an acceptable level of performance.

Topic Last Updated: v1.06 / March 10, 2015

e-Dif Rover Mode Operation

Rover mode operation of the Crescent receiver unit with the optional e-Dif application requires NMEA 0183 commands. These commands may be automatically issued through customized software or through a simple terminal interface running on a PC, PDA or data logger. See e-Dif Commands for detailed information on the commands supported by the e-Dif feature.

Topic Last Updated: v1.00 / August 11, 2010

e-Dif Startup

On startup, the receiver with the e-Dif application software running requires a few minutes to gather enough satellite tracking information to model the errors for the future. And in some environments this can take up to 10 minutes. The receiver does not have to be stationary for this process but it must be tracking the satellites throughout it. This process of gathering information and the subsequent initialization of e-Dif is referred to as "calibration."

Topic Last Updated: v1.00 / August 11, 2010

e-Dif Rover Calibration

Rover calibration is the process of modeling the errors at the rover. Calibration can be performed in either a relative or an absolute sense, depending on positioning needs. Relative positioning provides positions that are accurate to one another but there may be some offset from the true geographical position. Additionally, unless the same point is used for all calibrations and its assumed position stored, it is possible for different cycles of e-Dif to have an offset.

Calibrating for relative positioning is easier than for absolute position, since you are not restricted to using a point with known coordinates. Calibrating for absolute positioning mode requires placing the GPS antenna at a known reference location. Use this point for subsequent calibrations.

Topic Last Updated: v1.00 / August 11, 2010

e-Dif Rover Performance

The Crescent receiver's positioning performance is dependant upon the rate at which the environmental modeling of e-Dif and the environmental errors diverge. The more that e-Dif is able to model the errors correctly, the longer it will provide reliable and accurate positioning. As there is no way in real time to know the rate of divergence, a rule of thumb is to set the maximum age of differential to either 30 or 40 minutes, depending on how much error the application is able to tolerate (or simply recalibrate before 30 to 40 minutes goes by). Hemisphere GNSS testing has shown that relative accuracy will often be better than 1.0 m 95% of the time after 30 minutes of e-Dif operation.

You should perform testing at your location to determine the level of performance that would be seen on average. When testing this feature, it is a good idea to look at a number of e-Dif cycles per day, and monitor performance against a known coordinate and possibly other receivers in autonomous and differential mode. You should do this over a number of days with different states of the ionosphere.

You can monitor the energy level of the ionosphere based upon the amount of solar flare activityat http://www.spaceweather.com.

Topic Last Updated: v1.06 / March 10, 2015

L-Dif

L-Dif Local Differential Option

Local differential (L-Dif) is a specialized message type that can be sent only between two Crescent-based receivers. One receiver is used as the base station and must remain stationary. It is extremely useful to know the coordinates of the base station position but averaging the position over several days will also suffice. The second receiver is used as a rover and the messages must be sent either through a cable or over a radio link.

Topic Last Updated: v1.00 / August 11, 2010

L-Dif Startup

On startup, the receiver with the L-Dif running requires several commands to initialize the proprietary messages that are sent over the air.

Topic Last Updated: v1.00 / August 11, 2010

L-Dif Performance

The receiver's positioning performance in L-Dif mode is dependant upon:

- Environment of the base and rover receivers
- Distance between them and
- Accuracy of the entered coordinates of the base station

Hemisphere GNSS suggests you perform your own testing at your location to determine the level of performance you would expect on average. When testing this feature, conduct tests of 12-24 hours—in different environments—and monitor performance against a known coordinate. Do this over a number of days with different states of the ionosphere.

You can monitor the energy level of the ionosphere based upon the amount of solar flare activity at http://www.spaceweather.com.

Topic Last Updated: v1.06 / March 10, 2015

RTK Overview

Real Time Kinematic (RTK) positioning is the highest form of navigational accuracy for GNSS receivers. Hemisphere GNSS offers RTK for both Crescent and Eclipse platforms. See RTK commands for more information.

Topic Last Updated: v1.07 / February 16, 2017

Multi-Funcation Application (MFA) Software

Your device may include MFA software that allows you to set the positioning (mode) hierarchy of your device. To verify if your device contains MFA software send the $JAPP command to the device; the response indicates whether you have MFA as follows:

- Without MFA (two specific applications listed)
  Example: `$>JAPP,WAASRTKB,AUTODIFF,1,2`
- With MFA (MFA and one specific application listed)
  Example: `$>JAPP,MFA,SBASRTKB,1,2`

The hierarchy is the path your device follows to determine what differential source to use depending on available sources. The hierarchy is as follows:

1. RTK
2. L-band (Atlas)
3. SBAS
4. Beacon
5. External RTCM
6. Autonomous

If you are running RTK and you lose your RTK radio link, the device defaults to the next highest mode, being either Atlas high precision service or SBAS (if available). If the new signal becomes unusable, the next mode will be selected (for example Beacon or External RTCM). Finally, if no correction signals are available, the device defaults to Autonomous.

You can include or exclude specific sources. For example, you can exclude sources that you do not want your device to use, such as if you want to use only beacon. If you do not exclude the other sources your device may use SBAS instead. Another example is if you want to exclude Atlas (when you do not have an Atlas subscription) to conserve power. You include and exclude sources using
the $JDIFFX,INCLUDE and $JDIFFX,EXCLUDE commands, respectively.

Topic Last Updated: v1.07 / February 16, 2017

Post-Processing

Crescent and Eclipse receiver modules can output raw measurement data for post processing applications. The raw measurement and ephemeris data are contained in the following messages, which must be logged in a binary file:

- Observations: Bin 76 (GPS), Bin 66 (GLONASS), Bin 36 (BEIDOU)

  Or

  Bin 16 (All constellations; required for GALILEO)

- Ephemeris: Bin 95 (GPS), Bin 65 (GLONASS), Bin 35 (BEIDOU), Bin 45 (GALILEO)
- Time conversion: Bin 94 (GPS), Bin 34 (BEIDOU), Bin 44 (GALILEO)

(Crescent receivers must log Bin 94, 95, and 96 messages for GPS). Depending on the application, the binary data can be logged to a file and then translated to RINEX at a later time on a PC.

Hemisphere GNSS provides a RINEX translator. It is available by contacting technical support at Hemisphere GNSS; however, because there is limited ability to store station information in the binary file, developers may consider writing their own translator. Some code is available for developers but with very limited support. The code should be self-evident to developers familiar with RINEX and knowledgeable in C language.

Topic Last Updated: v1.07 / February 16, 2017

Hemisphere GNSS Hardware Platforms

Hardware Platforms Overview

Hemisphere GNSS offers the following hardware platforms:

- Crescent
- Crescent Vector II
- Eclipse II

Topic Last Updated: v1.06 / March 10, 2015

Universal Development Kit

The Universal Development Kit allows you to integrate a Hemisphere GNSS OEM board into your design and includes the following:

- Enclosure
- Main carrier board
- Set of three adapter boards for use with small form factor Hemisphere GNSS OEM boards
- Power cable and AC power supply
- Two serial cables - one straight serial cable and one null modem cable for RTK
- The Universal Development Kit supports the following Hemisphere GNSS OEM boards:
  - Enclosure
  - Crescent
  - Crescent Vector II
  - Eclipse II
  - miniEclipse
  - LX-2 (L-band DGPS and high precision services)

Depending on the Hemisphere GNSS OEM board you purchase with your Universal Development Kit, an Integrator's Guide is available for download from the Hemisphere GNSS website at www.hemispheregnss.com (search for Universal Development Kit).

Last Updated: v1.06 / March 10, 2015

Evaluating Receiver Performance

Hemisphere GNSS evaluates performance of the receiver with the objective of determining best-case performance in a real-world environment. Our testing has shown that the receiver achieves a performance better than 0.6 m 95% of the time in typical DGPS modes.

The qualifier of 95% is a statistical probability. Manufacturers often use a probability of RMS, one sigma, or one standard deviation. These three terms all mean the same thing and represent approximately 67% probability. Performance measures with these probabilities are not directly comparable to a 95% measure since they are lower probability (less than 70% probability).

Table 1 summarizes the common horizontal statistical probabilities.

Table 1: Horizontal Accuracy Probability Statistics

Accuracy Measure	Probability (%)
rms (root mean square)	63 to 68
CEP (circular error probability)	50
R95 (95% radius)	95 to 98
2drms (twice the distance root)	95

It is possible to convert from one statistic to another using Table 2. Using the value where the 'From' row meets the 'To' column, multiply the accuracy by this conversion value.

Table 2: Accuracy Conversions

	To			
From	CEP	rms	R95	2drms
CEP	1	1.2	2.1	2.4
rms	0.83	1	1.7	2.0
R95	0.48	.59	1	1.2
2drms	0.42	.5	.83	1

For example, Product A, after testing, has an accuracy of 90 cm 95% of the time (R95). To compare

this to Product B that has a sub-meter horizontal rms specification of 60 cm:

1. Select the value from where the 'R95' row and the 'rms' column intersect (to convert to rms). This conversion value is 0.59.

2. Multiply the 90 cm accuracy by this conversion factor and the result is 53 cm rms. Compared to Product B’s 60 cm specification of sub-meter rms, Product A offers better performance.

To properly evaluate one receiver against another statistically, the receivers should be using identical correction input (from an external source) and share the same antenna using a power splitter (equipped with appropriate DC-blocking of the receivers and a bias-T to externally power the antenna). With this setup, the errors in the system are identical with the exception of receiver noise.

Although this is a comparison of the GNSS performance qualities of a receiver, it excludes other performance merits of a GNSS engine. The dynamic ability of a receiver should always be compared in a similar way with the test subjects sharing the same antenna. Unless a receiver is moving, its software filters are not stressed in a similar manner to the final product application. When testing dynamically, a much more accurate reference would need to be used, such as an RTK system, so that a "truth" position per epoch is available.

Further, there are other performance merits of a GNSS engine such as its ability to maintain a lock on GNSS and SBAS satellites. When evaluating this ability, the same GNSS antenna should be shared between the receivers test subjects. For the sake of comparing the tracking availability of one receiver to another, no accurate "truth" system is required unless performance testing is also to be analyzed. Again, an RTK system would be required; however, it is questionable how its performance will fare with environments where there are numerous obstructions such as foliage. Other methods of providing a truth reference may need to be provided through observation times on surveyed monuments or traversing well-known routes.

Should you look to compare two RTK systems, determining truth can be very complicated. A rigorous dynamic comparison of two competing RTK systems should only be attempted by individuals and organizations familiar with RTK and potentially with inertial navigation equipment. Fortunately, most manufacturer's RTK performance is specified in similar accuracy values, and in general, RTK accuracy is quite similar across different manufacturers.

Note: Contact Hemisphere GNSS Technical Support for further assistance in developing a test setup or procedure for evaluation of the receiver.

Topic Last Updated: v1.07 / February 16, 2017

Receiver Operation

Receiver Operation Overview

When turned on, the receiver goes through an internal startup sequence. It is, however, ready to communicate immediately. Refer to the receiver-specific manual for the power specifications of the product.

When its antenna has an unobstructed view of the sky, the receiver provides a position in approximately 60 seconds and acquires SBAS lock in about 30 seconds more.

Note: The receiver can take up to 5 minutes to receive a full SBAS ionospheric map. Optimum accuracy is obtained when the receiver is processing corrected positions using complete ionosphere information.

Topic Last Updated: v1.00 / August 11, 2010

Communicating with the Receiver

Communicating with Receivers

The receiver module features three primary serial ports (A, B, C) that may be configured independently of each other.

The ports can be configured to output a combination of data types:

- NMEA 0183
- Hemisphere GPS proprietary binary format
- RTCM SC-104

The usual data output is NMEA 0183 messages because these are the industry standard.

Note: If different data types are required to be output from the receiver simultaneously, such as NMEA 0183 and binary or NMEA 0183 and RTCM SC-104, ensure that the software used for logging and processing of the data has been designed to correctly parse the different data types from the single stream of data.

Topic Last Updated: v1.00 / August 11, 2010

NMEA 0183 Messages

NMEA 0183 is a communications standard established by the National Marine Electronics Association (NMEA). NMEA 0183 provides data definitions for a variety of navigation instruments and related equipment such as gyrocompasses, Loran receivers, echo sounders, and GNSS receivers.

NMEA 0183 functionality is virtually standard on all GNSS equipment available. NMEA 0183 has an ASCII character format that enables the user to read the data via a receiving device with terminal software.

The following is an example of one second of NMEA 0183 data from the receiver:

```
$GPGGA,144049.0,5100.1325,N,11402.2729,W,1,07,1.0,1027.4,M,0,M,,010
*61
$GPVTG,308.88,T,308.88,M,0,0.04,N,0.08,K*42

$GPGSV,3,1,10,02,73,087,54,04,00,172,39,07,66,202,54,08,23,147,48,*7
9

$GPGSV,3,2,10,09,23,308,54,11,26,055,54,15,00,017,45,21,02,353,45*78

$GPGSV,3,3,10,26,29,257,51,27,10,147,45,45,,,,,,,,*74
```

The NMEA 0183 standard allows manufacturers to define proprietary custom commands and to combine data into proprietary custom messages. Proprietary NMEA 0813 messages are likely to be supported only by specific manufacturers.

All messages and ports can be configured independently (see example below).

Port	Baud Rate	Messages
A	9600	GPGGA, one every 1 second GPGSV, one every 5 seconds
B	19200	GPGGA, one every 2 seconds Bin1, one every 1 second Bin2, one every 1 second

A selection of NMEA 0183 data messages can be configured at various update rates with each message having a maximum update rate. A different selection of NMEA 0183 messages with different rates can be configured on another port.

Commands and Messages Overview presents information about the NMEA 0183 interface of the receiver smart antenna. See Reference Documents for contact information if you need to purchase a copy of the NMEA 0183 standard.

Topic Last Updated: v1.07 / February 16, 2017

Hemisphere GNSS Proprietary Binary Interface

Hemisphere GNSS proprietary binary messages may be output from the receiver simultaneously with NMEA 0183 messages.

Binary messages are inherently more efficient than NMEA 0183 and would be used when maximum communication efficiency is required. Some receiver-specific pieces of information are only available through binary messages, such as raw data for post processing.

Note: If you need to log binary data, make sure the logging software has opened the file as a binary file; otherwise, data may be lost.

Topic Last Updated: v1.06 / March 10, 2015

RTCM SC-104 Protocol

RTCM SC-104 is a standard that defines the data structure for differential correction information for a variety of differential correction applications. It was developed by the Radio Technical Commission for Maritime services (RTCM) and has become an industry standard for communication of correction information. RTCM is a binary data protocol and is not readable with a terminal program. Because it is a binary format and not ASCII text, it appears as "garbage" data on screen.

The following is an example of how the RTCM data appears on screen:


```
mRMP@PJfeUtNsmMFM{nVtIOTDbA^xGh~kDH`_FdW_yqLRryrDuh
cB\@}N`ozbSD@O^}nrGqkeTlpLLrYpDqAsrLRrQN{zW|uW@H`z]~aG
xWYt@I`_FxW_qqLRryrDCikA\@Cj]DE]|E@w_mIroMNjkKOsmMFM{ WDw
W@HVEbA^xGhLJQH`_F`W_aNsmMFM[WVLA\@S}amz@iIIuP
qx~IZhTCpLLrYpdP@kOsmMFM[kVDHwVGbA^P{WWuNt_SW_yMs
mMnqdrhcC\@sE^ZfC@}vJmNGAHJVhTCqLRryrdviStW@H_GbA^ P{wxu[k
```


All Hemisphere GNSS receivers support RTCM v2.x Type 1, Type 5, Type 6, and Type 9 messages for DGPS positioning.

Hemisphere GNSS receivers do not support RTCM v2.x messages for RTK positioning. However RTCM v3.x messages (Type 1001 through 1008) are suitable for RTK positioning.

Note: RTCM v2.x is a local area data standard. This means that performance degrades as a function of distance from the base station when:

- Positioning with external connection input to the receiver from an external source or
- Outputting corrections from the receiver to another GNSS receiver.

The additional degradation depends on the difference in observed orbit and ionospheric errors between the reference station and the remote unit. A general rule of thumb is an additional 1 m error per 100 miles.

This error is often seen as a bias in positioning, resulting in a position offset. The scatter of the receiver is likely to remain close to constant.

See Reference Documents for RTCM contact information to purchase a copy of the RTCM SC-104 specifications.

Topic Last Updated: v1.07 / February 16, 2017

Firmware and Subscriptions

Firmware

About Firmware

Hemisphere GNSS products are built on one of three receiver platforms, each of which has specific firmware applications available.

- Crescent - WAAS, e-Dif, Atlas service, L-Dif/RTK base, L-Dif/RTK rover
- Crescent Vector - WAAS, RTK rover
- Eclipse - WAAS/RTK base, RTK roverAtlas high precision services

Some products may require purchasing a subscription code to unlock specific functionality. See Subscription Codes for more information.

As its name suggests, firmware is somewhere between hardware and software. Like software, it is a computer program which is executed by a microprocessor or a microcontroller. But it is also tightly linked to a piece of hardware, and has little meaning outside of it.

Within the context of GNSS, the hardware is the GNSS receiver and it is the receiver's processor that executes the firmware. The receiver's processor supports two simultaneous versions of firmware but only one version operates at a given time. The two versions—referred to as applications—may have different functionality.

Use the JAPP command to change between two receiver applications.

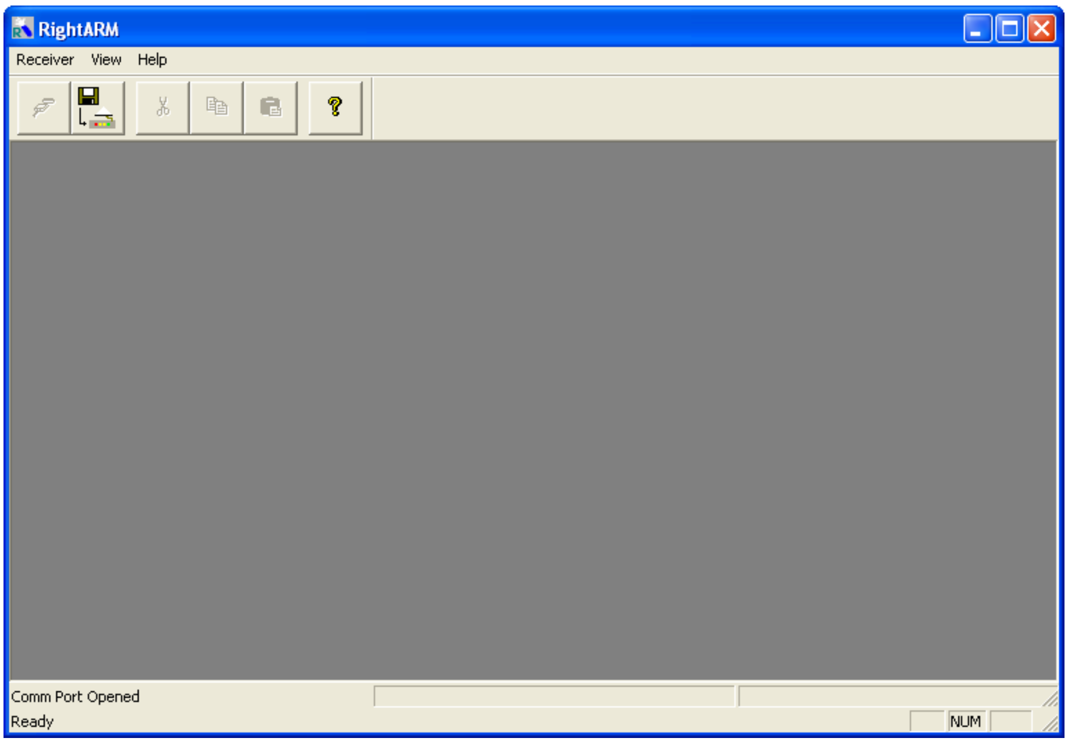
Topic last updated: v1.07/ February 16, 2017

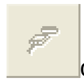
Using RightARM to Load Firmware

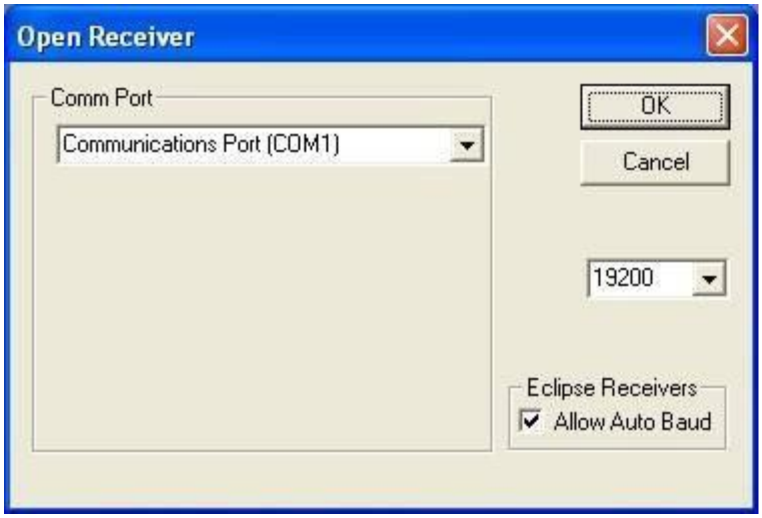
RightARM is Hemisphere GNSS software that allows you to load the various GNSS receiver firmware options and updates as they are provided by Hemisphere GNSS.

To load the firmware:

1. Download the latest version of RightARM from http://www.hemispheregnss.com.
2. Install RightARM application on your computer.
3. Connect the receiver to your computer and power on the receiver.
4. Double-click the RightARM icon  to launch the program. The following screen appears.



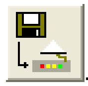
5. Click the **Open Receiver** button  or select **Receiver > Connect**. The Open Receiver window appears, so you can identify a connected receiver.

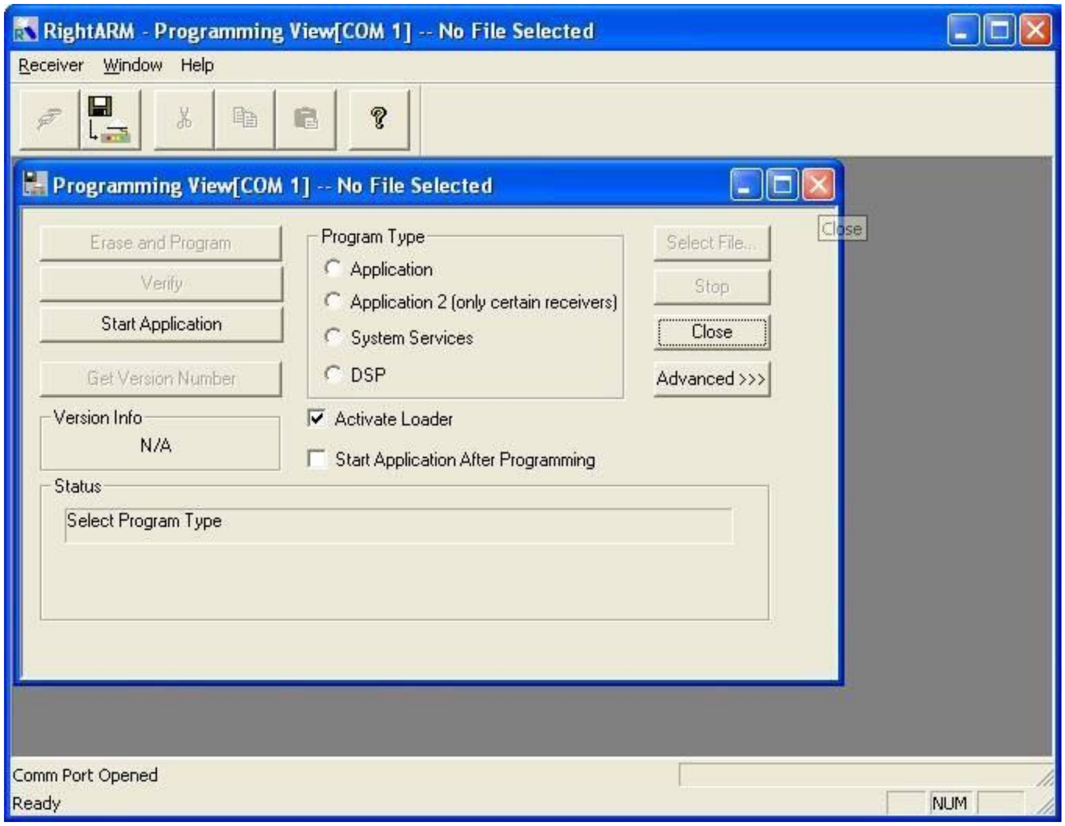


6. Select the **Comm Port** on your computer to which the receiver is connected, select the 19200 baud rate for the receiver, and then click **OK**.

Note: You must set the baud rate to 19200.

When RightARM has successfully connected to the receiver the following message appears in the lower left corner of the screen.

7. Click the **Programming View** button . The Programming View window appears, enabling you to select different firmware programming options.



8. Select the **Program Type** you want to install and then click **Select File**. The Open window appears.

Note: Most Hemisphere GNSS receivers have two application locations available for firmware. In this case, select the Application option under Program Type and follow the remaining steps. Once the process is complete, you will repeat the process, selecting the Application 2 option when you reach this step again.

9. Select the required firmware file from the location where you saved it on your computer and click Open**. "File Loaded" appears in the status window on the Programming View window.**

10. Click the **Erase and Program** button to erase the firmware that is currently installed on the receiver in the selected application location and install the newly selected file in its place. "Erasing...Please Wait" appears in the Status field and a progress bar below this message indicates the programming progress. Once the new firmware has been successfully loaded on the receiver "Programming Done" appears in the Status field.

Note: Before pressing the Erase and Program button, the Activate Loader check box in the Programming View window will be selected. After pressing the Erase and Program button, the check box should be cleared and the Status field should show that the receiver is in loader mode and ready to receive the new firmware file. If the Activate Loader check box remains selected, turn the receiver off and then back on again, close and restart RightARM, and then start over at step 5.

WARNING: **Do not to interrupt the power supply to the receiver, and do not interrupt the communication link between the PC and the receiver until programming is complete. Failure to do so may cause the receiver to become inoperable and will require it to be returned to the factory for repair.**

11. Once the appropriate firmware has been loaded, click the **Close** button to close the Programming View window.

Note: If a second application needs to be loaded, turn off the receiver, repeating all the steps starting at step 4, and on step 8 select the Application 2 option from the Program Type field.

12. Exit RightARM, turn off your receiver, and then disconnect the receiver from your computer.

Topic Last Updated: v1.07/ February 16, 2017

Subscriptions Codes

This section covers:

- Finding the serial number and inputting a subscription code (e-Dif, RTK, 20 Hz or 10Hz, etc.) into a Hemisphere GNSS receiver
- Viewing the status and interpreting the $JI subscription date codes
- The difference between the receiver's response to the $JK and $JI commands

Topic Last Updated: v1.07/ February 16, 2017

Subscribing to an Application

Activating an application code on a Hemisphere GNSS receiver requires the following:

- Serial communication cable to connect the Hemisphere GNSS receiver to the serial COM port on the computer
- Download SLXMon from the www.hemispheregnss.com and install it on your PC or use a generic terminal program such as HyperTerminal
- Load the application to which to subscribe onto the Hemisphere GNSS receiver (see Using RightARM to Load Firmware)
- Purchase the application subscription code from Hemisphere GNSS or an authorized Hemisphere GNSS representative

To activate the application on a Hemisphere GNSS receiver:

1. Connect the Hemisphere GNSS receiver to the serial COM port on the computer.
2. Start SLXMon.
3. Select **File > Connect** and then select the appropriate Comm Port and Baud Rate to open communication with the receiver.
4. Select **Control > View Command Page**.
5. In the Receiver Command Page window type **$JAPP** in the Message box and then click **Send**.
6. Confirm which applications are loaded onto the receiver and the order in which they appear in the Reply box.

Example Response (in Reply box):

```
$>JAPP,WAAS,DIFF
```

where WAAS (SBAS, EGNOS, MSAS) is the number one application (or application number 1) and DIFF (same as e-Dif) is the "other" application (or application number 2)

7. If DIFF is listed as application number 2 in the $JAPP response then type the following command in the Message box:

```
$JAPP,O
```

where 'O' is the "other" application in the example. This swaps the two applications so that DIFF is be the current application.

8. Type the following command in the Message box:

```
$JI
```

The first number in the response is the serial number of the receiver. Example

Response (in Reply box):

```
$>JI,810133,1,3,09031998,01/06/1998,12/31/2018,3.5,31
```

The serial number is 810133. You will need to provide it to Hemisphere GNSS with your request for an e-Dif subscription code.

9. Type the following command in the Message box after receiving the subscription code from Hemisphere GNSS:

```
$JK,nnnn
```

where 'nnnn' is the subscription number. The receiver will respond with "subscription accepted."

Topic Last Updated: v1.7 / February 16, 2017

Interpreting the $JI and $JK 'Date'/Subscription Codes

Subscriptions codes enable GNSS differential correction sources on your receiver. When discussing them it is important to understand the following.

- The YYYY component of a MM/DD/YYYY formatted date—returned by both the JI and JK commands—is not always just the year component of that date. When a date's year starts with 30, only the 30 represents the year - and that year is 3000. A subscription expiration date of 01/01/3000 effectively means there is no expiration date.

- The last two digits of the 30YY 'date' represent the data output rate (in Hz) and the GNSS differential correction sources that have been subscribed to and are therefore enabled on your receiver. Hemisphere GNSS refers to these two digits as the Additive Code (see Understanding Additive Codes).

- The 30 and the 00 in the 'year' 3000, then, represents "Expires 3000 (so effectively does not expire), the data rate is 10 Hz, and SBAS is enabled." The 'year' 3015 indicates "Expires 3000, the data rate is 20 Hz and differential correction sources SBAS/e-Dif/RTK and L-Dif have been subscribed to and are enabled."

Below is an example of the $JI command response, part of which is the subscription start and expiration dates (the Date Code is shaded).

```
$>JI,12838,1,7,26022003,01/01/1900,01/01/3000,6.8Hx,38
```

Topic Last Updated: v1.07 / February 16, 2017

Understanding Additive Codes

Tables 1 and 2 below provide subscription information for Crescent and Eclipse receivers, where the data rate and subscription are indicated by the 'date' returned by the JK and JI commands. For Eclipse II receivers, refer to Eclipse II Subscription Codes. The part of the date that indicates the data rate and subscription code is called the Additive Code. The last two digits in the subscription expiration date's 'year' comprise the Additive Codes, that is, the current data output rate from the receiver in Hz, plus the subscriptions—the enabled GPS differential correction sources.

Table 3 outlines the components of the Crescent, Eclipse, and Eclipse II Additive Codes. The subscription codes have different additive components for Crescent, Eclipse, and Eclipse II.

Table 1: Crescent Subscription Codes			
Date Code (Additive Code)	Hex Code	Maximum Data Rate	Subscription Description
3000 (0)	HEX 0	10 Hz	SBAS enabled
3001 (1)	HEX 1	20 Hz	SBAS enabled
3002 (0+2)	HEX 2	10 Hz	SBAS, e-Dif enabled
3003 (1+2)	HEX 3	20 Hz	SBAS, e-Dif enabled
3004 (0+4)	HEX 4	10 Hz	SBAS, RTK Rover enabled
3005 (1+4)	HEX 5	20 Hz	SBAS, RTK Rover enabled
3006 (0+2+4)	HEX 6	10 Hz	SBAS, RTK Rover, e-Dif enabled
3007 (1+2+4)	HEX 7	20 Hz	SBAS, RTK Rover, e-Dif enabled
3008 (0+8)	HEX 8	10 Hz	SBAS, L-Dif Rover, L-Dif Base, RTK Base enabled
3009 (1+8)	HEX 9	20 Hz	SBAS, L-Dif Rover, L-Dif Base, RTK Base enabled
3010 (0+2+8)	HEX A	10 Hz	SBAS, L-Dif Rover, L-Dif Base, RTK Base, e-Dif enabled
3011 (1+2+8)	HEX B	20 Hz	SBAS, L-Dif Rover, L-Dif Base, RTK Base, e-Dif enabled
3012 (0+4+8)	HEX C	10 Hz	SBAS, L-Dif Rover, L-Dif Base, RTK Rover, RTK Base enabled
3013 (1+4+8)	HEX D	20 Hz	SBAS, L-Dif Rover, L-Dif Base, RTK Rover, RTK Base enabled
3014 (0+2+4+8)	HEX E	10 Hz	SBAS, L-Dif Rover, L-Dif Base, RTK Rover, RTK Base, e-Dif enabled
3015 (1+2+4+8)	HEX F	20 Hz	SBAS, L-Dif Rover, L-Dif Base, RTK Rover, RTK Base, e-Dif enabled

Table 2: Eclipse Subscription Codes			
Date Code (Additive Code)	Hex Code	Maximum Data Rate	Subscription Description
3000 (0)	HEX 0	10 Hz	SBAS, Atlas enabled
3001 (1)	HEX 1	20 Hz	SBAS,Atlas enabled
3004 (0+4)	HEX 4	10 Hz	SBAS,Atlas , RTK Rover, RTK Base, Raw L1/L2 data enabled
3005 (1+4)	HEX 5	20 Hz	SBAS,Atlas , RTK Rover, RTK Base, Raw L1/L2 data enabled
3008 (0+8)	HEX 8	10 Hz	SBAS,Atlas , RTK Base, Raw L1/L2 data enabled
3009 (1+8)	HEX 9	20 Hz	SBAS,Atlas , RTK Base, Raw L1/L2 data enabled
3016 (0+16)	HEX 10	10 Hz	SBAS,Atlas , Raw L1/L2 data enabled
3017 (1+16)	HEX 11	20 Hz	SBAS,Atlas , Raw L1/L2 data enabled

Eclipse II Subscription Codes (go here)

Table 3: Crescent, Eclipse, and Eclipse II Additive Codes Components					
Crescent		Eclipse		Eclipse II	
Code	Description	Code	Description	Code	Description
0	10 Hz	0	10 Hz	0	10 Hz
1	20 Hz	1	20 Hz	1	20 Hz
2	e-Dif	2	n/a	2	e-Dif
4	L-Dif Rover, L-Dif Base, RTK Rover	4	Raw L1/L2 Data, RTK Base, RTK Rover	4	RTK Rover (minimum L1 only)
8	RTK Base	8	Raw L1/L2 Data, RTK Base	8	RTK Base (minimum L1 only)
16	n/a	16	Raw L1/L2 Data	16	Raw Data (minimum L1 only)
32	n/a	32	n/a	32	L2 signals
64	n/a	64	n/a	64	GLONASS signals (minimum L1 only)

Crescent Additive Code Examples

- 10 Hz (SBAS), e-Dif, and RTK is 0+2+4 = 6 (so 30**06**)
- 20 Hz (SBAS), e-Dif, and RTK is 1+2+4 = 7 (so 30**07**)

Comparing the JI and JK Responses

Example 1:

In the following Crescent examples, the Date Code is shaded.

- JI query date code example:

  `$>JI,311077,1,7,04102005,01/01/1900,01/01/3000,6.8Hx,46`

- JK query date code example:

  `$>JK,01/01/3000,0,(1, 2, 5 or no number)`

In the JK example the last two digits ('00') of the Date Code ('3000') represent the Hex Code (the second column of Table 2 above).

The last digit to the right (1, 2, 5 or no number) is the Downgrade Code...this is the output rate in Hertz indicating a downgrade from the default of 10 Hz. So if 1, 2 or 5 does not appear (no number), the output rate is the default 10 Hz.

The Date Codes are identical in either query and are directly related to each other. Also, the last digit in the JK query is the hexadecimal equivalent of the last two digits in the Date Code. The following example further illustrate this (Date Code is shaded).

Note: The JI response provides the decimal Date Code while the JK response provides both the decimal Date Code and the hex Date Code (the Hex Code).

Example 2:

`$>JI,311077,1,7,04102005,01/01/1900,01/01/3015,6.8Hx,46`

- JK query date code example:

  `$>JK,01/01/3015,F`

In this example the last two digits ('15') of the Date Code ('3015') is the decimal equivalent of the last value ('F'), which is the Hex Code (see the last row in Table 1 above). Example shows no downgrade code.

Topic Last Updated: v1.03 / January 11, 2012

Eclipse II Subscription Codes

Use the information below to determine your Eclipse II subscription code and its features.

	1	2	4	8	16	32	64		
	0x01	0x02	0x04	0x08	0x10	0x20	0x40		
	20Hz	e-Dif	RTK Rover, RTK Base, Raw Out	RTK Base, Raw Out	Raw Out	L2	GLONASS	Date Code (Additive Code)	Hex Code
Standard								3000	0
	Y							3001	1
		Y						3002	2
	Y	Y						3003	3
			Y					3004	4
	Y		Y					3005	5
		Y	Y					3006	6
	Y	Y	Y					3007	7
				Y				3008	8
	Y			Y				3009	9
		Y		Y				3010	A
	Y	Y		Y				3011	B
			Y	Y				3012	C
	Y		Y	Y				3013	D
		Y	Y	Y				3014	E
	Y	Y	Y	Y				3015	F
					Y			3016	10
	Y				Y			3017	11
		Y			Y			3018	12

Y	Y			Y		3019	13
		Y		Y		3020	14
Y		Y		Y		3021	15
	Y	Y		Y		3022	16
Y	Y	Y		Y		3023	17
			Y	Y		3024	18
Y			Y	Y		3025	19
	Y		Y	Y		3026	1A
Y	Y		Y	Y		3027	1B
		Y	Y	Y		3028	1C
Y		Y	Y	Y		3029	1D
	Y	Y	Y	Y		3030	1E
Y	Y	Y	Y	Y		3031	1F
					Y	3032	20
Y					Y	3033	21
	Y				Y	3034	22
Y	Y				Y	3035	23
		Y			Y	3036	24
Y		Y			Y	3037	25
	Y	Y			Y	3038	26
Y	Y	Y			Y	3039	27
			Y		Y	3040	28
Y			Y		Y	3041	29
	Y		Y		Y	3042	2A
Y	Y		Y		Y	3043	2B
		Y	Y		Y	3044	2C

Y		Y	Y		Y		3045	2D
	Y	Y	Y		Y		3046	2E
Y	Y	Y	Y		Y		3047	2F
				Y	Y		3048	30
Y				Y	Y		3049	31
	Y			Y	Y		3050	32
Y	Y			Y	Y		3051	33
		Y		Y	Y		3052	34
Y		Y		Y	Y		3053	35
	Y	Y		Y	Y		3054	36
Y	Y	Y		Y	Y		3055	37
			Y	Y	Y		3056	38
Y			Y	Y	Y		3057	39
	Y		Y	Y	Y		3058	3A
Y	Y		Y	Y	Y		3059	3B
		Y	Y	Y	Y		3060	3C
Y		Y	Y	Y	Y		3061	3D
	Y	Y	Y	Y	Y		3062	3E
Y	Y	Y	Y	Y	Y		3063	3F
						Y	3064	40
Y						Y	3065	41
	Y					Y	3066	42
Y	Y					Y	3067	43
		Y				Y	3068	44
Y		Y				Y	3069	45
	Y	Y				Y	3070	46

	Y	Y				Y	3071	47
Y			Y			Y	3072	48
			Y			Y	3073	49
Y	Y		Y			Y	3074	4A
	Y		Y			Y	3075	4B
Y		Y	Y			Y	3076	4C
		Y	Y			Y	3077	4D
Y	Y	Y	Y			Y	3078	4E
	Y	Y	Y			Y	3079	4F
Y				Y		Y	3080	50
				Y		Y	3081	51
Y	Y			Y		Y	3082	52
	Y			Y		Y	3083	53
Y		Y		Y		Y	3084	54
Y		Y		Y		Y	3085	55
	Y	Y		Y		Y	3086	56
Y	Y	Y		Y		Y	3087	57
			Y	Y		Y	3088	58
Y			Y	Y		Y	3089	59
	Y		Y	Y		Y	3090	5A
Y	Y		Y	Y		Y	3091	5B
		Y	Y	Y		Y	3092	5C
Y		Y	Y	Y		Y	3093	5D
	Y	Y	Y	Y		Y	3094	5E
Y	Y	Y	Y	Y		Y	3095	5F
					Y	Y	3096	60

Y					Y	Y	3097	61
	Y				Y	Y	3098	62
Y	Y				Y	Y	3099	63
		Y			Y	Y	3100	64
Y		Y			Y	Y	3101	65
	Y	Y			Y	Y	3102	66
Y	Y	Y			Y	Y	3103	67
			Y		Y	Y	3104	68
Y			Y		Y	Y	3105	69
	Y		Y		Y	Y	3106	6A
Y	Y		Y		Y	Y	3107	6B
		Y	Y		Y	Y	3108	6C
Y		Y	Y		Y	Y	3109	6D
	Y	Y	Y		Y	Y	3110	6E
Y	Y	Y	Y		Y	Y	3111	6F
				Y	Y	Y	3112	70
Y				Y	Y	Y	3113	71
	Y			Y	Y	Y	3114	72
Y	Y			Y	Y	Y	3115	73
		Y		Y	Y	Y	3116	74
Y		Y		Y	Y	Y	3117	75
	Y	Y		Y	Y	Y	3118	76
Y	Y	Y		Y	Y	Y	3119	77
			Y	Y	Y	Y	3120	78
Y			Y	Y	Y	Y	3121	79
	Y		Y	Y	Y	Y	3122	7A

75

Y	Y		Y	Y	Y	Y		3123	7B
		Y	Y	Y	Y	Y		3124	7C

Y		Y	Y	Y	Y	Y	3125	7D
	Y	Y	Y	Y	Y	Y	3126	7E
Y	Y	Y	Y	Y	Y	Y	3127	7F

Topic Last Updated: v1.03 / January 11, 2012

Determining the Receiver Type and Current Application

To determine the current receiver type, use the JT command. Table 1 shows the receiver type indicated by the JT response.

Table 1: $JT Response and Receiver Type

$JT Response	Receiver Type
SX1x	SX-1
SX2x	Crescent
SLXx	SLX2/SLX3
DF2x	Eclipse
DF3x	Eclipse II
MF3x	miniEclipse

The 'x' in the responses represents the receiver's current application. For example, if x = i, as in SX2i, 'i' is the application code for e-Dif.

Table 2 shows the application for the application code in the JT response.

Table 2: $JT Response and Application

$JT Responses with Application Code	Receiver Application
r	RTK rover
b	RTK base
i	e-Dif
g	L-band
g	WAAS
g	Standalone
a	Vector

Topic Last Updated: v1.02 / January 25, 2011

Configuring the Receiver

You can configure all aspects of receiver operation through any serial port using NMEA 0183 commands. You can:

- Select one of the two on-board applications
    - Two applications may be loaded at the same time, but only one can be active
    - You can select the active application through serial commands or through menu options on products with displays
- Set the baud rate of communication ports
- Select NMEA 0183 data messages to output on the serial ports and select the output rate of each message
- Set the maximum differential age cut-off
- Set the satellite elevation angle cut-off mask

The appropriate commands are described in Commands and Messages.

Topic Last Updated: v1.07 / February 16, 2017

Configuring the Data Message Output

In addition to its differential-only Port D, the receiver features three primary bidirectional ports referred to as A, B, and C. You can configure GPS data messages for all three ports by sending NMEA 0183 commands to the receiver module through all its communication ports. You can configure the output of Port B through A, for instance, and vice versa. The JASC NMEA message allows you to turn the messages on or off as required.

Note: For receivers that have a USB port that supports writing to a USB flash drive you can specify Port T as a port to receive messages.

In the examples below where you can specify the port, use 'PORTT' to specify Port T.

'THIS' Port and the 'OTHER' Port

The NMEA 0183 interface for Port A and B both use 'THIS' and 'OTHER' terminology.

- THIS port
  The port you are currently connected to for inputting commands. To get the data output through THIS port it is not necessary to specify 'this' (see Example 1 below).

- The OTHER port
  To specify the OTHER port for the data output, you need to include 'OTHER' in the command. See the two examples following which are both based on you being connected to Port B.

Example 1:

To turn the GPGGA message on at 5 Hz on Port B, use the following command:

$JASC,GPGGA,5<CR><LF>

Because B is THIS it does not have to be specified.

Example 2:

To turn the GPGGA message on at an output rate of 5 Hz on Port A, use the following command:

$JASC,GPGGA,5,OTHER<CR><LF>

Because B is THIS and A is OTHER, you have to specify OTHER. In contrast, when turning messages on or off on Port C from Port A or Port B, you must specify Port C in the command.

Example 3:

To turn the GPGLL NMEA 0183 message on at 10 Hz on Port C, use the following command:

$JASC,GPGLL,10,PORTC<CR><LF>

As with Port A and B, when communicating directly with Port C, you do not need to specify anything at the end of the message. See Commands and Messages for more information.

Topic Last Updated: v1.02 / January 25, 2011

Saving the Receiver Configuration

Each time the configuration of the receiver is changed, the new configuration should be saved so the receiver does not have to be reconsidered for the next power cycle.

To save the settings:

- Issue the JSAVE command. The receiver records the current configuration to non-volatile memory. The receiver indicates when the save process, which takes about five seconds, is complete.

Topic Last Updated: v1.00 / August 11, 2010

Using Port D for RTCM Input

The receiver has a port designed to accommodate externally supplied corrections input according to the RTCM SC-104 protocol. Port D provides this functionality although it has been fixed to operate at a baud rate of 9600 (8 data bits, no parity, and 1 stop bit, that is, 8-N-1).

To use Port D of the receiver for correction input, you must set the receiver to operate in beacon differential mode using the following command:

```
$JDIFF,BEACON<CR><LF>
```

This command was designed to “turn on” Port D differential operation in our products because many use the Hemisphere GNSS SBX beacon module interfaced to Port D.

Note: The receiver is compatible with RTCM SC-104 message types 1-3, 5-7, 9 and 16 although not all the message types contain differential data.

To return to using SBAS as the correction source, send the following command to the receiver:

```
$JDIFF,WAAS<CR><LF>
```

See Commands and Messages for detailed information on NMEA 0183 messages supported by the receiver.

Topic Last Updated: v1.06 / March 10, 2015

SBX-4 Database Mode

Enabling Database Mode

Database mode is automatically enabled when the SBX-4 receives a valid RMC message on Port 0. This requires the baud rate of Port 0 to be the same as the corresponding GPS receiver port.

Performance in Database Mode

In most installations Database mode will result in faster initial acquisition and better GPS accuracy compared to Auto mode.

In some installations Database mode may not work as well as Auto mode for the following reasons:

- The closest station is not in the station database and the SBX-4 has not yet received a Type7 Almanac message. Most stations now broadcast the Almanac message every ten minutes. Assuming the SBX-4 can tune to a surrounding station and receive a Type7 message, it will update the station database and automatically retune to the closest station.
- Signal quality in the area is poor. IEC61108-4 requires the receiver to switch away from a station when WER rises above 10%. For installations that do not need to comply with IEC61108-4 this threshold can be increased as usable corrections can be obtained for word error rates up to 50%.

Available Production Configuration Settings

Disable the automatic switch to Database mode:	`$PCSI,8,NITRAM,A`
Enable weak signal tracking (WER of 50%):	`$PCSI,8,NITRAM,W`
Enable legacy Q value output (in place of WER):	`$PCSI,8,NITRAM,Q`
Set SBX-4 to factory defaults:	`$PCSI,8,NITRAM,E`

Topic Last Updated: v1.00 / August 11, 2010

Ethernet Configuration

As of firmware version V5.6.1, the Hemisphere P328 receiver board has Ethernet support. It is disabled by default, but may be enabled.

The P328 is connected to a carrier board or enclosure which connects the P328's Ethernet pins to a standard RJ-45 jack (with integrated magnetics as appropriate).

Enabling and Disabling Ethernet

The full current state of Ethernet configuration may be checked with the command "**$JETHERNET**". Doing this when Ethernet is disabled should give a result like the following:

```
$JETHERNET
$>JETHERNET,MAC,8C-B7-F7-F0-00-01
$>JETHERNET,MODE,OFF
$>JETHERNET,PORTI,OFF
$> Current Ethernet IP Address: None
```

To enable Ethernet, determine if the receiver is allowed to be assigned an IP address automatically via DHCP, or statically assigned. If you are unsure, please contact the administrator of the network you wish to connect it to.

To enable Ethernet support with a DHCP-assigned IP address, simply use the command

```
$JETHERNET,MODE,DHCP
```

The receiver will attempt to get an address from the DHCP server on the network. You should be able to see the current IP address reported by a "**$JETHERNET**" query change.

To enable Ethernet support with a statically assigned IP address, use the command

```
$JETHERNET,MODE,STATIC,ip,subnet,gateway,dns
```

where **ip**/**subnet**/**gateway**/**dns** are each replaced with the relevant IP address. The **gateway** and **dns** parameters are optional, and only useful for allowing outgoing connections from the P328, which are not currently supported anyway. An example command would be

```
$JETHERNET,MODE,STATIC,192.168.0.42,255.255.255.0
```

If one wishes to disable Ethernet use the command

```
$JETHERNET,MODE,OFF
```

With Ethernet enabled, one can access the receiver on Windows machnies via "HGNSSxx", where "xx" is the receiver's ESN. For example from the command line one could use the command,

```
ping HGNSS1234567
```

Enabling Ethernet Services

With Ethernet enabled, it should be possible to send an ICMP ping to the P328 receiver from a PC on the same network, if one wishes to test that. No actual services are enabled on Ethernet by default however though, so to make practical use of Ethernet support, one must also enable a service.

As of the writing of this document, the only Ethernet service implemented is the PORTI virtual serial port. Additional types of Ethernet services may be implemented in future firmware versions.

The PORTI virtual serial port allows a listening TCP port to be opened, which will act just like a local serial port of the receiver would. Only one TCP client may be connected at a time.

Important Note: Enabling "PORTI" on Ethernet should only be done with the P328 connected to a trusted network, since it gives full access to the receiver just as a local serial port would, and has no authentication or security mechanisms.

To enable the PORTI service, use the command

$JETHERNET,PORTI,port

where **port** is replaced with the TCP port number which one wishes to use. Any port in the range 1 to 65535 is allowable, but it is recommended one consider which TCP port numbers are typically reserved for various common protocols and avoid those port numbers.

To disable the PORTI service, use the command

$JETHERNET,PORTI,OFF

Topic Last Updated: v1.07 / February 16, 2017

Commands and Messages

Commands and Messages Overview

The receiver supports a selection of NMEA 0183 messages, proprietary messages that conform to NMEA 0183 standards, and Hemisphere GNSS proprietary binary messages. It is your decision as a systems designer whether or not to support a NMEA 0183-only software interface or a selection of both NMEA 0183 and binary messages.

All Crescent and Eclipse receivers are configured with NMEA 0183 commands and can output NMEA 0183 messages. In addition to NMEA 0183, some receivers can be configured using NMEA 2000 commands and can output NMEA 2000 messages.

Commands

- General operation and configuration commands
- GNSS commands
- SBAS commands
- e-Dif commands
- Vector commands and messages
- GLONASS commands and messages
- DGPS base station commands
- Local differential and RTK commands and messages
- Beacon receiver commands and messages
- L-band commands
- RAIM commands

Messages

- Data messages
- Binary messages
- NMEA 2000 CAN messages

Topic Last Updated: v1.07 / February 16, 2017

NMEA 0183 Message Format

NMEA 0183 messages (sentences) have the following format:

$XXYYY,ZZZ,ZZZ,ZZZ...*CC<CR><LF>

where:

Element	Description
$	Message header character
XX	NMEA 0183 talker field (GP = GPS, GL = GLONASS, GA = GALILEO, GB = BEIDOU, GN = All constellations)
YYY	Type of GPS NMEA 0183 message
ZZZ	Variable length message fields
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Null (empty) fields occur when there is no information for that field. You can use the JNP command to specify the number of decimal places output in the GPGGA and GPGLL messages.

What does <CR><LF> mean?

The literal translation means "Carriage Return, Line Feed." They are terms used in computer programming languages to describe the end of a line or string of text. If you are writing your own communication software for a receiver, see some of the examples below. If you are already using a program such as Hemisphere GNSS' PocketMax, when you click to send a command to the receiver, the program adds the carriage return and line feed to the end of the text string for you. If you are using HyperTerminal or other terminal software, typically the Enter key on your keyboard is set to send the <CR><LF> pair. You may need to define this in the setup section of the terminal software. Some software may treat the Enter key on your numeric keypad differently than the main Enter key in the main QWERTY section of the keyboard – use the main Enter key for best results.

Originally, the carriage return and line feed characters were for use with printers. The carriage return character would signal the printer to send the print head back to the left edge of the page on the current line of text. The line feed command instructed the printer to advance the paper one line. Today, electronics often use the carriage return and line feed instructions to signify the end of a string of text, prompting the device to process the string and execute the instructions sent in the text string.

Electronics use different ways to represent the <CR><LF> characters. In ASCII numbers, <CR> is represented as 13 in decimal, or 0D in hexadecimal. ASCII for <LF> is 10 decimal, or 0A hexadecimal. Some computer languages use different ways to represent <CR><LF>. Unix and C language can use "\x0D\x0A". C language can also use "\r\n" in some instances. Java may use CR+LF. In Unicode, carriage return is U+000D, and line feed is U+000A. It is advised to clearly understand how to send these characters if you are writing your own interface software.

Topic Last Updated: v1.07 February 16, 2017

Command/Query/Message Types

General Operation and Configuration Commands

The following table lists the commands related to the general operation and configuration of the receiver.

Command	Description
JAIR	Specify how the receiver will respond to the dynamics associated with airborne applications
JALT	Turn altitude aiding for the receiver on or off
JAPP	Specify or query receiver application firmware
JASC,D1	Set the RD1 diagnostic information message from the receiver to on or off
JASC,VIRTUAL	Configure the receiver to have RTCM data input on one port and output through the other (when using an external correction source)
JBAUD	Specify the baud rates of the receiver or query the current setting
JBIN	Enable the output of the various binary messages supported by the receiver
JCONN	Create a virtual circuit between the A and B ports to enable communication through the receiver to the device on the opposite port
JDIFF	Specify or query the differential mode of the receiver
JDIFF,AVAILABLE	Query the receiver for the differential types currently being received
JDIFFX,EXCLUDE	Specify the differential sources to be excluded from operating in a multi-diff application
JDIFFX,GNSSOUT	Specify GNSS output in correction formats or query the current setting
JDIFFX,INCLUDE	Specify the differential sources to be allowed to operate in a multi-diff application
JDIFFX,SOURCE	Query the receiver for the differential source
JDIFFX,TYPE	Query the receiver for the differential type
JEPHOUT,PERIODSEC	to allow ephemeris messages (95, 65, 35) to go out a rate other than when they change
JFLASH,DIR	Display the files on a USB flash drive
JFLASH,FILE,CLOSE	Close an open file on a USB flash drive
JFLASH,FILE,NAME	Open a specific file, append to a specific file, or display the file name of the open file on a USB flash drive
JFLASH,FILE,OPEN	Create and open a file with an automatically generated file name on a USB flash drive
JFLASH,FREESPACE	Display the free space in kilobytes (KB) on a USB flash drive
JFLASH,NOTIFY,CONNECT	Enable/disable the automatic response when a USB flash drive is inserted or removed (if port is not specified the response will be sent to the port that issued the command)
JFLASH,QUERYCONNECT	Manually verify if a USB flash drive is connected or disconnected

JFORCEAPP	Force an application to be used in a multi-application (MFA)
JI	Display receiver information, such as its serial number and firmware version
JK	Subscribe the receiver to various options, such as higher update rates, e-Dif (or base station capability) or L-Dif; or query for the current subscription expiration date when running Atlas application or the receiver subscription code when running all other applications
JK,SHOW	contain authorization information
JLIMIT	Set the threshold of estimated horizontal performance for which the DGPS position LED is illuminated or query the current setting
JMODE	Query receiver for status of JMODE settings
JMODE,BASE	Enable/disable base mode functionality or query the current setting
JMODE,FIXLOC	Set the receiver to not re-average (or re-average) its position or query the current setting
JMODE,FOREST	Turn the higher gain functionality (for tracking under canopy) on/off or query the current setting
JMODE,GLOFIX	Enable/disable use of RTCM v3 (RTK) GLONASS correctors
JMODE,GPSONLY	Set the receiver to use GPS data in the solution or query the current setting (if GLONASS is available, setting to YES will cause the receiver to only use GPS data)
JMODE,L1ONLY	Set the receiver to use L1 data even if L2 data is available or query the current setting
JMODE,MIXED	Include satellites that do not have differential corrections in the solution
JMODE,NULLNMEA	Enable/disable output of NULL fields in NMEA 0183 messages when no there is no fix (when position is lost)
JMODE,SBASNORTK	Disable/enable the use of SBAS ranging signals (carrier phase) in RTK
JMODE,SBASR	Enable/disable SBAS ranging
JMODE,STRICTRTK	Use this command to invoke stricter checks on whether RTK fix is declared. Forces float of RTK at 30 seconds of Age-of-Diff
JMODE,SURETRACK	Enable/disable SureTrack functionality (default is enabled) or query the current setting
JMODE,SURVEY	Assure RTK fix is not declared when residual errors exceed 10 cm. Also forces use of GLONASS and prevents SureTrack operation.
JMODE,TIMEKEEP	Enable/disable continuous time updating in NMEA 0183 messages when there is no fix (when position is lost)
JMODE,TUNNEL	Enable/disable faster reacquisition after coming out of a tunnel or query the current setting
JPOS	Speed up the initial acquisition when changing continents with the receiver or query the receiver for the current position of the receiver

JPPS,FREQ	Specify the pps frequency of the receiver or query the current setting
JPPS,WIDTH	Specify the pps width of the receiver or query the current setting
JPRN,EXCLUDE	**For advanced users only.** Exclude GPS and/or other GNSS satellites from being used in the positioning solution or query the current setting
JQUERY,GUIDE	Query the receiver for its determination on whether or not it is providing suitable accuracy after both the SBAS and GPS have been acquired (up to five minutes)
JQUERY,TEMPERATURE	Query the receiver's temperature
JRELAY	Send user-defined text out of a serial port
JRESET	Reset the receiver to its default operating parameters by turning off outputs on all ports, saving the configuration, and setting the configuration to its defaults
JSAVE	Send this command after making changes to the operating mode of the receiver
JSHOW	Query the current operating configuration of the receiver
JSHOW,ASC	Query receiver for current ASCII messages being output
JSHOW,BIN	Query receiver for current Bin messages being output
JSHOW,CONF	Query receiver for configuration settings
JSHOW,GP	Query the receiver for each GP message currently being output through the current port and the update rate for that message
JSHOW,THISPORT	Query to determine which receiver port you are connected to
JSYSVER	Returns the boot loader version from the GPS card
JT	Query the receiver for its GPS engine type

Note: **Use the JSAVE command to save changes you need to keep and wait for the $>SAVE COMPLETE response.**

Topic Last Updated: v1.07 / February 16, 2017

GNSS Commands

The following table lists the commands supported by the internal GNSS engine for its configuration and operation.

Command	Description
JAGE	Specify maximum DGPS (COAST) correction age (6 to 8100 seconds)
JASC,GN	Enable the GPS data messages at a particular update rate to be turned on or off
JMASK	Specify the elevation cutoff mask angle for the GPS engine
JNMEA,PRECISION	Specify or query the number of decimal places to output in the GPGGA and the GPGLL messages or query the current setting
JNP	Specify the number of decimal places output in the GPGGA and GPGLL messages
JOFF	Turn off all data messages being output through the current port or other port
JOFF,ALL	Turn off all data messages being output through all ports
JSMOOTH	Set the carrier smoothing interval (15 to 6000 seconds) or query the current setting
JTAU,COG	Set the course over ground (COG) time constant (0.00 to 3600.00 seconds) or query the current setting
JTAU,SPEED	Set the speed time constant (0.00 to 3600.00 seconds) or query the current setting

Note: Use the JSAVE command to save changes you need to keep and wait for the $>SAVE COMPLETE response.

The following table lists the messages applicable to GNSS

Message	Description
Bin16	GNSS code and phase observation information
Bin19	GNSS Tracking Information

Topic Last Updated: v1.07/ February 16, 2017

SBAS Commands

The following table lists the commands supported by the SBAS demodulator for its control and operation.

Command	Description
JASC,D1	Set the RD1 diagnostic information message from the receiver to on or off
JASC,RTCM	Configure the receiver to output RTCM version 2 DGPS corrections from SBAS or beacon through either receiver serial port
JGEO	Display information related to the current frequency of SBAS and its location in relation to the receiver’s antenna
JWAASPRN	Change the SBAS PRNs in memory or query the receiver for current PRNs in memory

Note: Use the JSAVE command to save changes you need to keep and wait for the $>SAVE COMPLETE response.

Topic Last Updated: v1.00 / August 11, 2010

e-Dif Commands

The following table lists the commands supported by the e-Dif application for its control and operation.

Command	Description
JRAD,1	Display the current reference position in e-Dif applications only
JRAD,1,LAT,LON,HEIGHT	Use this command—a derivative of the JRAD,1,P command—when absolute positioning is required in e-Dif applications only
JRAD,1,P	e-Dif: Record the current position as the reference with which to compute e-Dif corrections. This would be used in relative mode as no absolute point information is specified. DGPS Base Station: Record the current position as the reference with which to compute Base Station corrections in e-Dif applications only. This would be used in relative mode as no absolute point information is specified
JRAD,2	Forces the receiver to use the new reference point (you normally use this command following a JRAD,1 type command)
JRAD,3	Invoke the e-Dif function once the unit has started up with the e-Dif application active, or, update the e-Dif solution (calibration) using the current position as opposed to the reference position used by the JRAD,2 command
JRAD,7	Turn auto recalibration on or off

Note: Use the JSAVE command to save changes you need to keep and wait for the $>SAVE COMPLETE response.

Topic Last Updated: v1.02 / January 25, 2011

Vector Commands and Messages

The following table lists the commands related to the GPS heading aspect of the Vector OEM heading system.

Command	Description
JASC	Turn on different messages
JASC,INTLT	Configure the receiver to output pitch and roll data (pitch and roll are factory calibrated over temperature to be accurate to ±3°C)
JASC,PASHR	Configure the receiver to output time, true heading, roll, and pitch data in one message
JASC,PTSS1	Configure the receiver to output heave, pitch, and roll in the commonly used TSS1 message format
JATT,COGTAU	Set the course over ground (COG) time constant (0.0 to 3600.0 seconds) or query the current setting
JATT,CSEP	Query for the current separation between GPS antennas
JATT,EXACT	Enable/disable internal filter reliance on the entered antenna separation or query the current setting
JATT,FLIPBRD	Turn the flip feature on/off (allowing you to install the Crescent Vector board upside down) or query the current feature status
JATT,GYROAID	Turn gyro aiding on or off or query the current setting
JATT,HBIAS	Set the heading bias or query the current setting
JATT,HELP	Show the available commands for GPS heading operation and status
JATT,HIGHMP	Set/query the high multipath setting for use in poor GPS environments
JATT,HRTAU	Set the heading rate time constant or query the current setting
JATT,HTAU	Set the heading time constant or query the current setting
JATT,LEVEL	Turn level operation on or off or query the current setting
JATT,MOVEBASE	Set the auto GPS antenna separation or query the current setting
JATT,MSEP	Manually set the GPS antenna separation or query the current setting
JATT,NEGTILT	Turn the negative tilt feature on or off or query the current setting
JATT,NMEAHE	Instruct the Crescent Vector to preface the HDG, HDM, HDT, and ROT messages with GPor HE
JATT,PBIAS	Set the pitch/roll bias or query the current setting
JATT,PTAU	Set the pitch time constant or query the current setting
JATT,ROLL	Configure the Crescent Vector for roll or pitch GPS antenna orientation
JATT,SEARCH	Force the Crescent Vector to reject the current GPS heading solution and begin a new search
JATT,SPDTAU	Set the speed time constant (0.0 to 3600.0 seconds) or query the current setting
JATT,SUMMARY	Display a summary of the current Crescent Vector settings
JATT,TILTAID	Turn tilt aiding on or off or query the current setting
JATT,TILTCAL	Calibrate tilt aiding or query the current feature status

The following table lists Vector messages.

Message	Description
GNGSA	GNSS DOP and active satellites
GPDTM	Datum reference
GPGGA	GPS fix data
GPGLL	Geographic position - latitude/longitude
GPGNS	GNSS fix data
GPGRS	GNSS range residuals
GPGST	GNSS pseudorange error statistics
GPGSV	GNSS satellite in view
GPHDG/HEHDG	Provide magnetic deviation and variation for calculating magnetic or true heading
GPHDM/HEHDM	Provide magnetic heading of the vessel derived from the true heading calculated
GPHDT/HEHDT	Provide true heading of the vessel
GPHEV	Heave value in meters
GPRMC	Recommended minimum specific GNSS data
GPROT/HEROT	Contains the vessel's rate of turn (ROT) information
GPRRE	Range residual message
GPVTG	Course over ground and ground speed
GPZDA	Time and date
PASHR	Time, true heading, roll, and pitch data in one message
PSAT,GBS	Satellite fault detection used for RAIM
PSAT,HPR	Proprietary NMEA sentence that provides the true heading, pitch/roll information and time ina single message
PSAT,INTLT	Proprietary NMEA sentence that provides the title measurement from the internal inclinometer (in degrees)
TSS1	Heave, pitch, and roll message in the commonly used TSS1 message format

Topic Last Updated: v1.07 / Octoter 13, 2016

GLONASS Commands and Messages

The following table lists the commands applicable to GLONASS-capable receivers.

Command	Description
JASC,GL	Enable the GLONASS data messages at a particular update rate to be turned on or off. When turning messages on, various update rates are available depending on the requirements.
JMODE,GPSONLY	Set the receiver to use GPS data in the solution or query the current setting (if GLONASS is available, setting to YES will cause the receiver to only use GPS data)
JNMEA,GGAALLGNSS	Configure the GGA string to include full GNSS information (the number of used GLONASS satellites will be included in the GPGGA message) or query the current setting

The following table lists the messages applicable to GLONASS-capable receivers.

Message	Description
Bin16	GALILEO GNSS code and phase observation information
Bin62	GLONASS almanac information
Bin65	GLONASS ephemeris information
Bin66	GLONASS L1 code and carrier phase information
Bin69	GLONASS L1 diagnostic information
GLMLA	GLONASS almanac data - contains complete almanac data for one GLONASS satellite (multiple sentences may be transmitted, one for each satellite in the GLONASS constellation)

Topic Last Updated: v1.02 / January 25, 2011

GALILEO Commands and Messages

The following table lists the commands applicable to GALILEO-capable receivers.

Command	Description
JASC,GAGSV	Enable/disable the data for GALILEO satellites in view. When turning messages on, various update rates are available depending on the requirements.
JASC,GNGNS	Enable/disable fix data for GNSS systems including GALILEO (GAGNS). When turning messages on, various update rates are available depending on the requirements.
JMODE,GPSONLY	Set the receiver to use GPS data in the solution or query the current setting (if GALILEO is available, setting to YES will cause the receiver to only use GPS data)
JNMEA,GGAALLGNSS	Configure the GGA string to include full GNSS information (the number of used satellites will be included in the GPGGA message) or query the current setting

The following table lists the messages applicable to GALILEO-capable receivers.

Message	Description
Bin45	GALILEO ephemeris information
Bin16	GALILEO GNSS code and phase observation information
Bin44	GALILEO time conversion information

***Note: For observations in tracking status, see GNSS, Bin 16 & Bin 19.**

Topic Last Updated: v1.07 / February 16, 2017

QZSS Commands and Messages

The following table lists the commands applicable to QZSS-capable receivers.

Command	Description
JASC,GQGSV	Enable/disable the data for QZSS satellites in view.
JASC,GNGNS	Enable/disable fix data for GNSS systems.
JASC,GNGSA	DOP and active satellite information

The following table lists the binary messages applicable to QZSS-capable receivers.

Message	Description
Bin16	GNSS code and phase observation information
Bin19	GNSS diagnostic information

Topic Last Updated: v1.07 / February 16, 2017

DGPS Base Station Commands

The following table lists the commands supported by the base station feature for its control and operation.

Command	Description
JRAD,1	Display the current reference position in e-Dif applications only
JRAD,1,LAT,LON,HEIGHT	Use this command—a derivative of the JRAD,1,P command—when absolute positioning is required in e-Dif applications only
JRAD,1,P	**e-Dif:** Record the current position as the reference with which to compute e-Dif corrections. This would be used in relative mode as no absolute point information is specified. **DGPS Base Station:** Record the current position as the reference with which to compute Base Station corrections in e-Dif applications only. This would be used in relative mode as no absolute point information is specified
JRAD,9	Initialize the Base Station feature and use the previously entered point, either with $JRAD,1,P or $JRAD,1,LAT,LON,HEIGHT, as the reference with which to compute Base Station corrections in e-Dif applications only. Use this for both relative mode and absolute mode.
JRAD,10	Specify BDS message to be transmitted by base station

Topic Last Updated: v1.02 / January 25, 2011

Local Differential and RTK Commands and Messages

The following table lists the commands supported by Local Differential (L-Dif) and RTK feature for its control and operation.

Command	Description
JASC,CMR	Set the proprietary CMR messages to on or off to provide corrections to the rover (only applies to an Eclipse base station receiver when using GPS dual frequency RTK mode)
JASC,DFX	Set the proprietary DFX messages to on or off to provide corrections to the rover (only applies to a Crescent base receiver when using L-Dif or RTK mode)
JASC,ROX	Set the proprietary ROX messages to on or off to provide corrections to the rover (only applies to an Eclipse base station receiver when using GPS dual frequency RTK mode)
JASC,RTCM3	Set the RTCM version 3 messages to on or off to provide corrections to the rover (only applies to an Eclipse base station receiver when using GPS dual frequency RTK mode)
JASC,PSAT,BLV,1	Configure the receiver to output the North,East,Up base-line vector
JASC,PSAT,FVI,1	Configure the receiver to output a message include most position and attitude information
JASC,PSAT,RTKPROG	Configure the receiver to output RTK fix progress
JASC,PSAT,RTKSTAT	Configure the receiver to output the most relevant parameters affecting RTK
JASC,PSAT,VCT,1	Configure the receiver to output the heading, pitch, roll, and master to slave vector
JMODE,BASE	Enable/disable base mode functionality or query the current setting
JNMEA,PRECISION	Specify or query the number of decimal places to output in the GPGGA andthe GPGLL messages or query the current setting
JNP	Specify the number of decimal places output in the GPGGA and GPGLLmessages
JQUERY,RTKPROG	Perform a one-time query of RTK fix progress information
JQUERY,RTKSTAT	Perform a one-time query of the most relevant parameters that affect RTK
JRTK,1	Show the receiver's reference position (can issue command to base station or rover)
JRTK,1,LAT,LON,HEIGHT	Set the receiver's reference position to the coordinates you enter (canissue command to base station or rover)
JRTK,1,P	Set the receiver's reference coordinates to the current calculated position if you do not have known coordinates for your antenna location (can issue command to base station or rover)
JRTK,5	Show the base station's transmission status for RTK applications (can issue command to base station)
JRTK,5,Transmit	Suspend or resume the transmission of RTK (can issue command to base station)
JRTK,6	Display the progress of the base station (can issue command to base station)
JRTK,12	Disable or enable the receiver to go into fixed integer mode (RTK) vs. float mode (L-Dif) - can issue command to rover
JRTK,17	Display the transmitted latitude, longitude, and height of the base station (can issue command to base station or rover)
JRTK,18	Display the distance from the rover to the base station, in meters (can issue command to rover)
JRTK,18,BEARING	Display the bearing from the base station to the rover, in degrees (can issue command to rover)
JRTK,18,NEU	Display the distance from the rover to the base station and the delta North, East, and Up, in meters (can issue command to rover)
JRTK,28	Set the base station ID transmitted in ROX/DFX/CMR/RTCM3 messages (can issue command to base station)

JRTCM3, ANTNAME	Specify the antenna name that is transmitted in various RTCM3 messages from the base
JRTCM3, EXCLUDE	Specify RTCM3 message types to not be transmitted (excluded) by base station
JRTCM3, INCLUDE	Specify RTCM3 message types to be transmitted by base station
JRTCM3, NULLANT	Specify the antenna name as null (no name) that is transmitted in various RTCM3 messages from the base

The following table lists the Local Differential (L-Dif) and RTK messages.

Message	Description
PSAT,RTKPROG	Contains RTK fix progress information
PSAT,RTKSTAT	Contains the most relevant parameters affecting RTK

Topic Last Updated: v1.07 / Octoter 13, 2016

Beacon Receiver Commands and Messages

If integrating a Hemisphere GNSS SBX beacon module with the receiver GNSS engine, Hemisphere GNSS recommends interfacing the beacon receiver to Port D of the receiver engine. Hemisphere GNSS has implemented some command and message pass-through intelligence for such an integration. In this configuration you can issue the commands in the following table to the beacon receiver through either Port A, Port B, or Port C of the receiver. When you issue queries to the SBX primary communications port, the response messages are output interspersed with RTCM correction information. This may cause conflicts with a GNSS receiver's ability to compute differential corrected solutions. By sending these queries to the SBX secondary communications port the flow of RTCM corrections on the primary port will not be interrupted.

The following table lists the beacon commands/messages found in this Help file.

Query	NMEA 0183 Query Type	Description
GPCRQ,MSK	Standard	Query the SBX for its operational status
GPCRQ,MSS	Standard	Query the SBX for its performance status
GPMSK	Standard	Tune beacon the receiver and turn on diagnostic information
PCSI,0	Hemisphere GNSS proprietary	Query the SBX to output a list of available proprietary PCSI commands
PCSI,1	Hemisphere GNSS proprietary	Query the SBX for a selection of parameters related to the operational status of its primary channel
PCSI,1,1	Hemisphere GNSS proprietary	Obtain beacon status information from the SBX beacon engine inside the receiver
PCSI,2	Hemisphere GNSS proprietary	Query the SBX to output a selection of parameters related to the operational status of its secondary channel
PCSI,3,1	Hemisphere GNSS proprietary	Query the SBX to output the search information used for beacon selection in Automatic Beacon Search mode. The output has three frequencies per line.
PCSI,3,2	Hemisphere GNSS proprietary	Display the ten closest beacon stations
PCSI,3,3	Hemisphere GNSS proprietary	Display the contents of the beacon station database
PCSI,4	Hemisphere GNSS proprietary	Clear search history in Auto mode
PCSI,5	Hemisphere GNSS proprietary	Set the baud rate of Port0 and Port1
PCSI,6	Hemisphere GNSS proprietary	Reboot SBX receiver
PCSI,7	Hemisphere GNSS proprietary	Swap modes on the receiver

The following table lists the beacon messages found in this Help file.

Message	Description
CRMSK	Operational status message of SBX
CRMSS	Performance status message of SBX

Topic Last Updated: v1.06 / March 10, 2015

Atlas® Commands

The following tables lists the commands accepted by the Atlas-band receiver to configure and monitor the Atlas functionality of the receiver.

Command	Description
$JI	Requests the serial number and firmware version number from the receiver
$JK	Requests the authorization from the receiver Is used to send the authorization to the receiver
$JASC,GPGGA,1	Requests receiver to output GGA positions at 1Hz.
$JASC,RD1,1	EnablesAtlas Diagnostic message output
$JDIFF,LBAND,SAVE	EnablesAtlas mode for tracking the Atlas communication satellites
$JDIFF,INCLUDE,ATLAS	Enables the Atlas solution in the receiver
$JFREQ,AUTO	Automatically sets theAtlas parameters to track the Atlas communication satellites
$JSAVE	Saves issued commands

Note: Use the JSAVE command to save changes you need to keep and wait for the $J>SAVE COMPLETE response.

If your Atlas communication is working properly the following should apply:

- **Bit Error Rate**: less than 10-10
- **Spot Beam Freq**:
  - **AMERICAS**: 1545.5300
  - **APAC**: 1539.8525
  - **EMEA**: 1540.9525
- **Nav Condition**: FFFFF

If this is not the case, then enter the following commands in the Receiver Command Page, one at a time:

Command
$JFREQ,AUTO
$JDIFF,LBAND,SAVE
$JFREQ,AUTO
$JDIFF,LBAND,SAVE

Topic Last Updated: v1.07 / February 16, 2017

RAIM Commands

RAIM (Receiver Autonomous Integrity Monitoring) is a GNSS integrity monitoring scheme that uses redundant ranging signals to detect a satellite malfunction resulting in a large range error. The Hemisphere GNSS products use RAIM to alert users when errors have exceeded a user-specified tolerance. RAIM is available for SBAS, and Beacon, applications.

The following table lists the available RAIM commands.

Command	Description
JRAIM	Specify the parameters of the RAIM scheme that affect the output of the PSAT,GBS message or query the current setting

Topic Last Updated: v1.07 / February 16, 2017

Data Messages

Note: 20 Hz output is only available with a 20 Hz subscription.

Message	Maximum Rate	Description
GNGSA	1 Hz	GPS DOP and active satellite information
GPALM	1 Hz	GPS almanac data
GPGGA	20 Hz	Detailed GPS position information
GPGLL	20 Hz	Latitude and longitude data
GPGNS	20 Hz	Fixes data for single or combined satellite navigation systems
GPGRS	20 Hz	Supports Receiver Autonomous Integrity Monitoring (RAIM)
GPGST	1 Hz	GNSS pseudorange error statistics
GPGSV	1 Hz	GNSS satellite in view
GPHDG/HEHDG	20 Hz	Magnetic deviation and variation for calculating magnetic or true heading
GPHDM/HEHDM	20 Hz	Magnetic heading of the vessel derived from the true heading calculated
GPHDT/HEHDT	20 Hz	True heading of the vessel
GPHEV	20 Hz	Heave value in meters
GPRMC	20 Hz	Recommended minimum specific GNSS data
GPROT/HEROT	20 Hz	Vessel's rate of turn (ROT) information
GPRRE	1 Hz	Range residual message
GPVTG	20 Hz	Course over ground and ground speed
GPZDA	20 Hz	UTC time and date information
PASHR	1 Hz	Time, true heading, roll, and pitch data in one message
PSAT,ATTSTAT	1HZ	
PSAT,GBS	1 Hz	Used to support Receiver Autonomous Integrity Monitoring (RAIM)
PSAT,HPR	20 Hz	Proprietary NMEA message that provides the true heading, pitch, roll, and time in a single message
PSAT,INTLT	1 Hz	Proprietary NMEA message that provides the tilt measurements from the internal inclinometers (in degrees)
PSAT,RTKPROG	1 Hz	Contains RTK fix progress information
PSAT,RTKSTAT	1 Hz	Contains the most relevant parameters affecting RTK

RD1	1 Hz	SBAS diagnostic information
TSS1	20 Hz	Heave, pitch, and roll message in the commonly used TSS1 message format

Topic Last Updated: v1.07 / February 16, 2017

Binary Messages

Message Structure

The binary messages supported by the receiver are in an Intel Little Endian format for direct read in a PC environment. More information on this format at the following web site:

http://www.cs.umass.edu/~verts/cs32/endian.html

Each binary message begins with an 8-byte header and ends with a carriage return, line feed pair (0x0D, 0x0A). The first four characters of the header is the ASCII sequence $BIN.

The following table provides the general binary message structure.

Component	Description	Type	Bytes	Values
Header	Synchronization String	4 byte string	4	$BIN
	Block ID - type of binary message	Unsigned short	2	1, 2, 80, 93, 94, 95, 96, 97, 98, or 99
	DataLength - the length of the binary messages	Unsigned short	2	52, 16, 40, 56, 96, 128, 300, 28, 68, or 304
Data	Binary Data - varying fields of data with a total length of DataLength bytes	Mixed fields	52, 16, 40, 56, 96, 128, 300, 28, 68, or 304	Varies - see message tables
Epilogue	Checksum - sum of all bytes of the data (all DataLength bytes); the sum is placed in a 2-byte integer	Unsigned short	2	Sum of data bytes
	CR- Carriage return	Byte	1	0D hex
	LF - Line feed	Byte	1	0A hex

Messages

Message	Description
Bin1	GPS position message (position and velocity data)
Bin2	GPS DOPs (Dilution of Precision)
Bin3	Lat/Lon/Hgt, Covariances, RMS, DOPs and COG, Speed, Heading
Bin16	All constellation code and phase information
Bin 19	GNSS diagnostic information
Bin35	BeiDou ephemeris information
Bin36	BeiDou code and carrier phase information (all frequencies)
Bin44	GALILEO time conversion
Bin45	GALILEO ephemeris
Bin62	GLONASS almanac information
Bin65	GLONASS ephemeris information
Bin66	GLONASS L1/L2 code and carrier phase information

Bin69	GLONASS L1/L2 diagnostic information
Bin76	GPS L1/L2 code and carrier phase information
Bin80	SBAS data frame information
Bin89	SBAS satellite tracking information
Bin93	SBAS ephemeris information
Bin94	Ionospheric and UTC conversion parameters
Bin95	GPS ephemeris information
Bin96	GPS L1 code and carrier phase information
Bin97	Processor statistics
Bin98	GPS satellite and almanac information
Bin99	GPS L1 diagnostic information
Bin 100	GPS L2 diagnostic information
Bin209	SNR and status for all GNSS tracks

Topic Last Updated: v1.07 / February 16, 2017

NMEA 2000 CAN Messages

Message	Description
GNSSPositionData	Detailed GPS position information
GNSSPositionRapidUpdates	Abbreviated GPS position information
NMEACogSogData	GPS speed and direction information

Topic Last Updated: v1.00 / August 11, 2010

GPCRQ

GPCRQ,MSK Command

Command Type Beacon Receiver

Description Standard NMEA 0183 query to prompt the SBX for its operational status (response is the CRMSK message)

You can issue this command through the secondary serial port with a standard response issued to the same port. This will not affect the output of RTCM data from the main serial port when the receiver has acquired a lock on a beacon station.

Command Format `$GPCRQ,MSK<CR><LF>`

Receiver Response `$CRMSK,fff.f,X,ddd,Y,n*CC<CR><LF>`

where

Response Component	Description
fff.f	Frequency in kHz (283.5 to 325)
X	Tune mode (M = manual, A = automatic, D = database)
ddd	MSK bit rate (100 or 200 bps)
Y	MSK rate selection mode (M = manual, A = automatic, D = database)
n	Period of output of CRMSS performance status message (0 to 100 seconds)

Example Response example:
`$CRMSK,322.0,M,100,A,2*CC`

The frequency is 322.0 kHz, tune mode is Manual, MSK bit rate is 100 bps, MSK rate selection mode is Automatic, and the message is output every 2 seconds.

Additional Information

Topic Last Updated: v1.04 / May 29, 2012

GPCRQ,MSS Command

Command Type

Beacon Receiver

Description

Standard NMEA 0183 query to prompt the SBX for its performance status (response is the CRMSS message)

You can issue this command through the secondary serial port with a standard response issued to the same port. This will not affect the output of RTCM data from the main serial port when the receiver has acquired a lock on a beacon station.

Command Format

`$GPCRQ,MSS<CR><LF>`

Receiver Response

`$CRMSS,xx,yy,fff.f,ddd*CC<CR><LF>`

where

Response Component	Description
xx	Signal strength in dBµV/m
yy	Signal-to-noise ratio (SNR) in dB
fff.f	Frequency in kHz (283.5 to 325)
ddd	MSK bit rate in bps (100 or 200)

Example

Response example:
`$CRMSS,65,36,322.0,100*CC`

The signal strength is 65 dBµV/m, SNR is 36 dB, frequency is 322.0 kHz, and MSK bit rate is 100 bps.

Additional Information

Topic Last Updated: v1.04 / May 29, 2012

GPMSK Command

Command Type [Beacon Receiver]

Description

Beacon Tune command

Instruct the SBX to tune to a specified frequency and automatically select the correct MSK rate. When you send this command through Port A, Port B, or Port C, it is automatically routed to Port D. The resulting confirmation of this message is returned to the same port from which you sent the command.

Command Format

`$GPMSK,fff.f,F,mmm,M[,n]<CR><LF>`

where:

Command/Response Component	Description
fff.f	Beacon frequency in kHz (283.5 to 325) This may be left blank if the following field 'F' is set to 'A' (automatic) or 'D' (database)
F	Frequency selection mode (M = manual, A = automatic, D = database)
mmm	MSK bit rate This may be left blank if the following field 'M' is set to 'A' (automatic) or 'D' (database)
M	MSK rate selection mode (M = manual, A = automatic, D = database)
n	Period of output of CRMSS performance status message (0 to 100 seconds), where leaving the field blank will output the message once **Note:** This field is optional when using database tuning mode or automatic tuning mode.

Receiver Response

`$CRMSS,xx,yy,fff.f,ddd*CC<CR><LF>`

where

Response Component	Description
xx	Signal strength in dBμV/m
yy	Signal-to-noise ratio (SNR) in dB
fff.f	Frequency in kHz (283.5 to 325)
ddd	MSK bit rate in bps (100 or 200)

Example

To instruct the SBX to tune to 310.5 kHz with a bit rate of 100 and output the CRMSS message every 20 seconds issue the following command:

`$GPMSK,310.5,M,100,M,20<CR><LF>`

...and the receiver response is:

```
$CRMSS,65,36,310.5,100*CC
```

(repeating every n=20 seconds)

If using database tuning mode issue the following command:

```
$GPMSK,,D,,D<CR><LF>
```

If using automatic tuning mode issue the following command:

```
$GPMSK,,A,,A<CR><LF>
```

Additional Information	When the SBX acknowledges this message, it immediately tunes to the specified frequency and demodulates at the specified rate. When you set 'n' to a non-zero value, the SBX outputs the CRMSS message at that period through the serial port from which the SBX was tuned. When you issue this command with a non-zero 'n' value through Port B, the periodic output of the CRMSS performance status message does not impact the output of RTCM on Port A. However, when tuning the SBX with a non-zero 'n' value through Port A, the CRMSS message is interspersed with the RTCM data. Most GPS engines will not be able to filter the CRMSS message, causing the overall data to fail parity checking. When power to the SBX is removed and reapplied, the status output interval resets to zero (no output). When tuning the SBX engine, if the 'n' field in this message is non-zero, the CRMSS message output by the SBX may interrupt the flow of RTCM data to the GPS receiver. Repower the SBX to stop the output of the CRMSS message or retune the Beacon receiver with 'n' set to zero.

Topic Last Updated: v1.02 / January 25, 2011

JAGE Command

Command Type	GPS
Description	Specify maximum DGPS (COAST) correction age (6 to 8100 seconds). Using COAST technology, the receiver can use old correction data for extended periods of time. The default setting for the receiver is 2700 seconds. If you select a maximum correction age older than 1800 seconds (30 minutes), test the receiver to ensure the new setting meets the requirements, as accuracy will slowly drift with increasing time.
Command Format	`$JAGE,age<CR><LF>` where 'age' is the maximum differential age timeout
Receiver Response	`$>`
Example	To set the DGPS correction age to 60 seconds issue the following command: `$JAGE,60<CR><LF>`
Additional Information	To query the receiver for the current DGPS correction age, issue the JSHOW command. What does <CR><LF> mean?

Topic Last Updated: v1.02 / January 25, 2011

JAIR Command

Command Type [General Operation and Configuration](#)

Description Specify how the receiver will respond to the dynamics associated with airborne applications or query the current setting

Command Format

Specify how the receiver responds

```
$JAIR,r<CR><LF>
```

where 'r' is the AIR mode:

- NORM - normal track and nav filter bandwidth
- HIGH - highest track and nav filter bandwidth (receiver is optimized for the high dynamic environment associated with airborne platforms)
- LOW - lowest track and nav filter bandwidth
- AUTO - default track and nav filter bandwidth, similar to NORM but automatically goes to HIGH above 30 m/sec

Query the current setting

```
$JAIR<CR><LF>
```

Receiver Response

Receiver response when specifying how the receiver responds or querying the current setting

```
$>JAIR,MAN,NORM
$>JAIR,MAN,HIGH
$>JAIR,MAN,LOW
$>JAIR,AUTO,NORM
```

Example To set the AIR mode to LOW issue the following command:

```
$JAIR,LOW<CR><LF>
```

The response is then:

```
$>JAIR,MAN,LOW<CR><LF>
```

Additional Information Defaults to normal (NORM) which is recommended for most applications. The AUTO option enables the receiver to decide when to turn JAIR to HIGH.

CAUTION: Setting AIR mode to HIGH is not recommended for Crescent Vector operation.

On the HIGH setting, the receiver tolerates larger and sudden drops in the SNR value before it discards the data as being invalid. This additional tolerance is beneficial in applications such as crop dusting where an aircraft is banking rapidly. As the aircraft banks, the antenna position shifts from upright and having a clear view of the sky to being tipped slightly, with a possibly obscured view of the sky, and then back to upright. This sudden tipping of the antenna causes the SNR value to drop.

If the tolerance is not set as HIGH, the receiver views the data recorded while banking as invalid and discards it. As a result the GPS position will not be accurate.

The status of this command is also output in the JSHOW message.

Topic Last Updated: v1.02 / January 25, 2011

JALT Command

Command Type [General Operation and Configuration](#)

Description Turn altitude aiding for the receiver on or off

When set to something other than NEVER, altitude aiding uses a fixed altitude instead of using one satellite's observations to calculate the altitude. The advantage of this feature, when operating in an application where a fixed altitude is acceptable, is that the extra satellite's observations can be used to the betterment of the latitude, longitude, and time offset calculations, resulting in improved accuracy and integrity. Marine markets, for example, may be well suited for use of this feature.

Command Format

`$JALT,c[,h[,GEOID]]<CR><LF>`

where 'c' (feature status variable) and 'h' (threshold variable) may be one of the following:

c Value	Corresponding h Value	Description	Format
NEVER	N/A	Default mode of operation where altitude aiding is not used.	`$JALT,NEVER<CR><LF>`
SOMETIMES	PDOP	Sets the receiver to use altitude aiding depending upon the PDOP threshold.	`$JALT,SOMETIMES,PDOP<CR><LF>`
SATS	NUMSATS	Sets the receiver to use altitude aiding depending upon the number of visible satellites. If there are fewer visible satellites than specified by NUMSATS, altitude aiding is used.	`$JALT,SATS,NUMSATS<CR><LF>`
ALWAYS	HEIGHT	Sets the receiver to use altitude aiding regardless	`$JALT,ALWAYS,HEIGHT<CR><LF>` `$JALT,ALWAYS,HEIGHT,GEOID<CR><LF>`

		of a variable. In this case, you may specify the ellipsoidal altitude HEIGHT that the receiver should use.	

To obtain a HEIGHT value to use with ALWAYS (using DGPS positions), average the HEIGHT over a period of time (the longer the time period, the more accurate this HEIGHT value). This is the ellipsoidal height.
$JALT,ALWAYS,HEIGHT<CR><LF>

If you use the height reported from the GPGGA message (this is actually geoidal and not ellipsoidal), use the following command:
$JALT,ALWAYS,HEIGHT,GEOID<CR><LF>

Receiver Response

$>

Example

To turn altitude aiding on to SOMETIMES with a PDOP of 5 issue the following command:
$JALT,SOMETIMES,5<CR><LF>
7
To turn altitude aiding on to ALWAYS using the height of 401.6 m as reported in the GPGGA message (geoidal height) issue the following command:
$JALT,ALWAYS,401.6,GEOID<CR><LF>

Additional Information

To query the receiver for the current setting, issue the JSHOW command. For example, if you issue the following command:

$JALT,ALWAYS,404.2<CR><LF>

...then issuing the JSHOW command displays the following as part of its output:

$>JSHOW,ALT,ALWAYS,404.2

Topic Last Updated: v1.03 / January 11, 2012

JAPP Command

Command Type	General Operation and Configuration
Description	Specify which of the installed applications should be utilized or query the receiver for the currently installed applications **Note:** Hemisphere GNSS Crescent and Eclipse GPS receivers are able to hold up to two different application firmware programs simultaneously.
Command Format	**Specify receiver application firmware (when two applications are present)** `$JAPP,OTHER<CR><LF> or $JAPP,O<CR><LF>` **(the second command uses the letter O, not a zero) or** `$JAPP,x<CR><LF>` **where 'x' is either 1 (application in slot 1) or 2 (application in slot 2)** **Query receiver application firmware** `$JAPP<CR><LF>`
Receiver Response	For example, if WAAS (SBAS) and AUTODIFF (e-Dif) are the two installed applications (WAAS in slot1 and AUTODIFF in slot2) and WAAS is the current application, if you issue the `$JAPP,OTHER<CR><LF>`command on a receiver, the response to `$JAPP<CR><LF>` will be`$>JAPP,AUTODIFF,WAAS,2,1,` indicating that application slot 2 (e-Dif) is currently being used. Hemisphere GNSS recommends that you follow up the sending of these commands with a `$JAPP` query to see which application is 1 or 2. It is best to use these two commands when upgrading the firmware inside the receiver, because the firmware upgrading utility uses the application number to designate which application to overwrite. Response to querying the current setting `$>JAPP,CURRENT,OTHER,[1 OR 2],[2 OR 1]` where: • 'CURRENT' indicates the current application in use • 'OTHER' indicates the secondary application that is not currently in use • 1 and 2 indicate in which application slots the applications reside

Example

If the response to $JAPP<CR><LF> is $>JAPP,WAAS,AUTODIFF,1,2, this indicates:

- WAAS (SBAS) is the current application and is in application slot 1
- e-Dif is the other application (not currently used) and is in application slot 2

Additional Information

When querying the current setting, the following application names may appear (depending on your product):

- Crescent

WAAS – Changes to the SBAS application. For the sake of the application names, the SBAS application is referred to as WAAS by the receiver's internal firmware

  - AUTODIFF – Changes to the e-Dif application. Referred to as "AUTODIFF" in the receiver's internal firmware
  - LOCRTK – Changes to the local differential rover application
  - RTKBAS – Changes to the local differential base application
  - LBAND – Changes toAtlas DGPS service

- Eclipse

WAASRTKB – Changes to the SBAS/RTK Base application

  - LBAND – Changes toAtlas DGPS service
  - RTK – Changes to the RTK Rover application

- Eclipse II

SBASRTKB – Changes to the SBAS/L-band/RTK Base application

  - AUTODIFF – Changes to the e-Dif application, referred to as "AUTODIFF" in the firmware
  - RTK – Changes to the RTK Rover application
  - MFA - Multi-function application

- miniEclipse

WAASRTKB – Changes to the SBAS/RTK Base application

  - AUTODIFF – Changes to the e-Dif application, referred to as "AUTODIFF" in the firmware
  - RTK – Changes to the RTK Rover application
  - MFA - Multi-function application

JASC

JASC Command Overview

The JASC command is used to request ASCII messages.

Command	Description
JASC,CMR	Set the proprietary CMR messages to on or off to provide corrections to the rover
JASC,D1 (RD1)	Set the RD1 diagnostic information message from the receiver to on or off
JASC,DFX	Set the proprietary DFX messages to on or off to provide corrections to the rover
JASC,GL	Enable the GLONASS data messages at a particular update rate to be turned on or off. When turning messages on, various update rates are available depending on the requirements.
JASC,GN	Enable the GNSS data messages at a particular update rate to be turned on or off. When turning messages on, various update rates are available depending on the requirements.
JASC,GP	Enable the GPS data messages at a particular update rate to be turned on or off
JASC,INTLT	Configure the receiver to output pitch and roll data
JASC,PASHR	Configure the receiver to output time, true heading, roll, and pitch data in one message
JASC,PSAT,ATTSTAT	Configure the receiver to output the information of secondary antenna
JASC,PSAT,BLV,1	Configure the receiver to output the North,East,Up base-line vector
JASC,PSAT,FVI,1	Configure the receiver to output a message include most position and attitude information
JASC,PSAT,RTKPROG	Configure the receiver to output RTK fix progress
JASC,PSAT,RTKSTAT	Configure the receiver to output the most relevant parameters affecting RTK
JASC,PSAT,VCT,1	Configure the receiver to output the heading, pitch, roll, and master to slave vector
JASC,PTSS1	Configure the receiver to output heave, pitch, and roll in the commonly used TSS1 message format
JASC,ROX	Set the proprietary ROX messages to on or off to provide corrections to the rover
JASC,RTCM	Configure the receiver to output RTCM version 2 DGPS corrections from SBAS or beacon through either receiver serial port
JASC,RTCM3	Set the RTCM version 3 messages to on or off to provide corrections to the rover
JASC,VIRTUAL	Configure the receiver to have RTCM data input on one port and output through the other (when using an external correction source)

Topic Last Updated: v1.07 / February 16, 2017

JASC,CMR Command

Command Type	Local Differential and RTK
Description	Set the proprietary CMR messages to on or off to provide corrections to the rover This command only applies to an Eclipse base station receiver when using GPS dual frequency RTK mode. RTK is relative to the reference position (base only).
Command Format	`$JASC,CMR,r[,OTHER]<CR><LF>` where: • 'r' = correction status variable (0 = turn corrections Off, 1 = turn corrections On) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.
Receiver Response	`$>`
Example	To turn on CMR messages on the OTHER port issue the following command: `$JASC,CMR,1,OTHER<CR><LF>`
Additional Information	To query the receiver for the current setting, issue the JSHOW command. To change the broadcast station ID, use JRTK,28.

Topic Last Updated: v1.02 / January 25, 2011

JASC,D1 Command

Command Type	General Operation and Configuration, SBAS
Description	Set the RD1 diagnostic information message from the receiver to on or off There is currently only an (R)D1 message.
Command Format	`$JASC,D1,r[,OTHER]<CR><LF>` where: • 'r' = message rate (0 = Off, 1 = On at 1Hz) • ',OTHER' = optional field, enacts a change in the RD1 message on the current port when you send the command without it (and without the brackets) and enacts a change in the RD1 message on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.
Receiver Response	`$>`
Example	To output the RD1 message once per second from THIS port issue the following command: `$JASC,D1,1<CR><LF>` ...and the output will look similar to the following: `$RD1,410213,1052,1551.489,1,0,39,- 611.5,0,1F,1F,0,999999` `$RD1,410214,1052,1551.489,1,0,40,-615.1,0,1F,1F,0,999999` `$RD1,410215,1052,1551.489,1,0,40,-607.1,0,1F,1F,0,999999` See RD1 message for a description of each field in the response.
Additional Information	Although you request D1 through this command the responding message is RD1. To query the receiver for the current setting, issue the JSHOW command. For example, if you issue the following command: `$JASC,D1,1<CR><LF>` ...then issuing the JSHOW command displays the following as part of its output:

```
$>JSHOW,ASC,D1,1
```

Topic Last Updated: v1.02 / January 25, 2011

JASC,DFX Command

Command Type	Local Differential and RTK
Description	Set the proprietary DFX messages to on or off to provide corrections to the rover This command only applies to a Crescent base receiver when using L-Dif or RTK mode. Differential is relative to the reference position (base only). See the JASC,ROX command for the equivalent message for the Eclipse series of products.
Command Format	`$JASC,DFX,r[,OTHER]<CR><LF>` where: • 'r' = correction status variable (0 = turn corrections Off, 1 = turn corrections On) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.
Receiver Response	$>
Example	To turn on DFX messages on THIS port issue the following command: `$JASC,DFX,1<CR><LF>`
Additional Information	To query the receiver for the current setting, issue the JSHOW command. To change the broadcast station ID, use JRTK,28.

Topic Last Updated: v1.02 / January 25, 2011

JASC,GL Command

Command Type GLONASS

Description Enable the GLONASS data messages at a particular update rate to be turned on or off. When turning messages on, various update rates are available depending on therequirements.

Command Format `$JASC,msg,r[,OTHER]<CR><LF>`

where:

- 'msg' = name of the data message
- 'r' = message rate (see table below)
- ',OTHER' = optional field, enacts a change on the current port (THIS port) whenyou send the command without it (and without the brackets) and enacts a change onthe other port (OTHER port) when you send the command with it (without thebrackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Send a command with a zero value for the 'R' field to turn off a message.

MSG	R (rate in Hz)	Description
GLMLA	1 (on) or 0 (off) When set to on the message is sent once (one message for each tracked satellite) and then sent again whenever satellite information changes	GLONASS almanac data
GLGGA	20, 10, 2, 1, 0 or .2	GPS fix data
GLGLL	20, 10, 2, 1, 0 or .2	Geographic position - latitude/longitude
GLGNS	20, 10, 2, 1, 0 or .2	GNSS fix data
GLGSA	1 or 0	GLONASS DOP and active satellites
GLGSV	1 or 0	GLONASS satellite in view

Receiver Response `$>`

Example To output the GLGNS message through the OTHER port at a rate of 20 Hz, issue the following command:
`$JASC,GLGNS,20,OTHER<CR><LF>`

Additional Information The status of this command is also output in the JSHOW message.
What does <CR><LF> mean?

Updated: v1.02 / January 25, 2011

JASC,GA Command

Command Type GALILEO

Description Enable the GALILEO data messages at a particular update rate to be turned on or off. When turning messages on, various update rates are available depending on the requirements.

Command Format

```
$JASC,msg,r[,OTHER]<CR><LF>
```

where:

- 'msg' = name of the data message
- 'r' = message rate (see table below)
- ',OTHER' = optional field, enacts a change on the current port (THIS port) when you send the command without it (and without the brackets) and enacts a change on the other port (OTHER port) when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Send a command with a zero value for the 'R' field to turn off a message.

MSG	R (rate in Hz)	Description
GNGNS	20, 10, 2, 1, 0 or .2	All GNSS fix data (GAGNS output is GALILEO)
GAGSV	1 or 0	GALILEO satellites in view

Receiver Response

```
$>
```

Example To output the GAGNS message through the OTHER port at a rate of 20 Hz, issue the following command:

Additional Information The status of this command is also output in the JSHOW message.
What does <CR><LF> mean?

Topic Last Updated: v1.07 / February 16, 2017

JASC,GQ Command

Command Type	QZSS
Description	Enable the QZSS data messages at a particular update rate to be turned on or off.

Command Format

`$JASC,msg,r[,OTHER]<CR><LF>`

where:

- 'msg' = name of the data message
- 'r' = message rate (see table below)
- ',OTHER' = optional field, enacts a change on the current port (THIS port) when you send the command without it (and without the brackets) and enacts a change on the other port (OTHER port) when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Send a command with a zero value for the 'R' field to turn off a message.

MSG	R (rate in Hz)	Description
GQGSV	1 or 0	QZSS satellites in view

Receiver Response

`$>`

Example

To output the GAGNS message through the OTHER port, issue the following command:
`$JASC,GNGNS,1,OTHER<CR><LF>`

Additional Information

The status of this command is also output in the JSHOW message.
What does <CR><LF> mean?

Topic Last Updated: v1.07 / February 16, 2017

JASC,GN Command

Command Type GPS, Vector

Description Enable the GNSS data messages at a particular update rate to be turned on or off. When turning messages on, various update rates are available depending on the requirements.

Command Format

```
$JASC,msg,r[,OTHER]<CR><LF>
```

where:

- 'msg' = name of the data message
- 'r' = message rate (see table below)
- ',OTHER' = optional field, enacts a change on the current port (THIS port) when you send the command without it (and without the brackets) and enacts a change on the other port (OTHER port) when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Send a command with a zero value for the 'R' field to turn off a message.

MSG	R (rate in Hz)	Description
GNGGA	20, 10, 2, 1, 0 or .2	GNSS fix data
GNGLL	20, 10, 2, 1, 0 or .2	Geographic position - latitude/longitude
GNGNS	20, 10, 2, 1, 0 or .2	GNSS fix data
GNGSA	1 or 0	GNSS DOP and active satellites

Receiver Response

```
$>
```

Example To output the GNGNS message through the OTHER port at a rate of 20 Hz, issue the following command:

```
$JASC,GNGNS,20,OTHER<CR><LF>
```

Additional Information The status of this command is also output in the JSHOW message.

What does <CR><LF> mean?

Topic Last Updated: v1.07 / February 16, 2017

JASC,GP Command

Command Type GPS, Vector

Description Enable the GPS data messages at a particular update rate to be turned on or off. When turning messages on, various update rates are available depending on the requirements.

Command Format

```
$JASC,msg,r[,OTHER]<CR><LF>
```

where:

- 'msg' = name of the data message
- 'r' = message rate (see table below)
- ',OTHER' = optional field, enacts a change on the current port (THIS port) when you send the command without it (and without the brackets) and enacts a change on the other port (OTHER port) when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Send a command with a zero value for the 'R' field to turn off a message.

MSG	R (rate in Hz)	Description
GPALM	1 or 0	GPS almanac data
GPDTM	1 or 0	Datum reference
GPGBS	1 or 0	Satellite fault detection used for RAIM
GPGGA	20, 10, 2, 1, 0 or .2	Detailed GPS position information
GPGLL	20, 10, 2, 1, 0 or .2	Latitude and longitude data
GPGNS	20, 10, 2, 1, 0 or .2	Fixes data for single or combined satellite navigation systems
GPGRS	1, 0 or .2	GNSS range residuals
GNGSA	1 or 0	GPS DOP and active satellite information
GPGST	1 or 0	GNSS pseudorange error statistics
GPGSV	1 or 0	GNSS satellite in view
GPHDG or HEHDG	20, 10, 2, 1, 0 or .2	Magnetic deviation and variation for calculating magnetic or true heading
GPHDM or HEHDM	20, 10, 2, 1, 0 or .2	Magnetic heading of the vessel derived from the true heading calculated
GPHDT or HEHDT	20, 10, 2, 1, 0 or .2	True heading of the vessel
GPHEV	20, 10, 2, 1, 0 or .2	Heave value in meters
GPHPR	20, 10, 2, 1, 0 or .2	Proprietary NMEA message that provides the true heading, pitch, roll, and time in a single message

GPRMC	10, 2, 1, 0 or .2	Recommended minimum specific GNSS data
GPROT or HEROT	20, 10, 2, 1, 0 or .2	Vessel’s rate of turn (ROT) information
GPRRE	1 or 0	Range residual message
GPVTG	20, 10, 2, 1, 0 or .2	Course over ground and ground speed
GPZDA	20, 10, 2, 1, 0 or .2	UTC time and date information
INTLT	1 or 0	Proprietary NMEA message that provides the tilt measurements from the internal inclinometers (in degrees)

Receiver Response	`$>`
Example	To output the GPGGA message through the OTHER port at a rate of 20 Hz, issue the following command: `$JASC,GPGGA,20,OTHER<CR><LF>`
Additional Information	The status of this command is also output in the JSHOW message. What does <CR><LF> mean?

Topic Last Updated: v1.06 / March 10, 2015

JASC,INTLT Command

Command Type	Vector
Description	Configure the receiver to output pitch and roll data (pitch and roll are factory calibrated over temperature to be accurate to ±3°C) Saved with JSAVE.
Command Format	`$JASC,INTLT,r[,OTHER]<CR><LF>` where: • 'r' = message rate (0 = Off, 1 = On at 1Hz) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.
Receiver Response	`$PSAT,INTLT,pitch,roll*CC<CR><LF>` where pitch and roll are in degrees
Example	
Additional Information	PSAT,INTLT message

Topic Last Updated: v1.06 / March 10, 2015

JASC,PASHR Command

Command Type Vector

Description Configure the receiver to output time, true heading, heave, roll, and pitch data in one message

Command Format

```
$JASC,PASHR,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate (0 = Off, 1 = On at 1Hz)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without th brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Receiver Response

```
$PASHR,hhmmss.ss,HHH.HH,T,RRR.RR,PPP.PP,heave,rr.rrr,pp.ppp,hh.hhh,QF*CC<CR>
<
```

where:

Message Component	Description
hhmmss.ss	UTC time
HHH.HH	Heading value in decimal degrees
T	True heading (T displayed if heading is relative to true north)
RRR.RR	Roll in decimal degrees (- sign will be displayed when applicable)
PPP.PP	Pitch in decimal degrees (- sign will be displayed when applicable)
heave	Heave, in meters
rr.rrr	Roll standard deviation in decimal degrees
pp.ppp	Pitch standard deviation in decimal degrees
hh.hhh	Heading standard deviation in decimal degrees
QF	Quality Flag • 0 = No position • 1 = All non-RTK fixed integer positions • 2 = RTK fixed integer position
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example To turn on the PASHR message on THIS port issue the following command:

```
$JASC,PASHR,1<CR><LF>
```

...and the message output appears similar to the following:

```
$PASHR,162930.00,,T,2.48,3.92,-0.64,0.514,0.514,0.000,1*05

$PASHR,162931.00,,T,2.38,3.93,-0.70,0.508,0.508,0.000,1*07

$PASHR,162932.00,,T,2.67,4.00,-0.66,0.503,0.503,0.000,1*04
```

Additional Information	PASHR message

Topic Last Updated: v1.06 / March 10, 2015

JASC,PSAT,ATTSTAT Command

Command Type	Local Differential and RTK
Description	The information of secondary antenna.
Command Format	`$JASC,PSAT,ATTSTAT,r[,OTHER]<CR><LF>` where: • 'r' = message rate (0 = Off, 1 = On at 1Hz) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology. .
Receiver Response	`$>`
Example	To turn on this message on the THIS port issue the following command: `$JASC,PSAT,ATTSTAT,1<CR><LF>`
Additional Information	Issuing the JSAVE command after setting JASC,PSAT,ATTSTAT to 1 (message on at 1Hz) does not save this setting. You must enable JASC,PSAT,ATTSTAT (set it to 1) each time you power on the receiver.
Related Commands and Messages	PSAT,ATTSTAT message

Topic Last Updated: v1.07 / Octoter 13, 2016

JASC,PSAT,BLV Command

Command Type	Local Differential and RTK
Description	Configure the receiver to output the North, East,Up base-line vector
Command Format	`$JASC,PSAT,BLV,r[,OTHER]<CR><LF>` where: • **'r' = message rate 0,1,2,5,10,20 (0 = Off, 1 = On at 1Hz)** • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology. .
Receiver Response	`$>`
Example	To turn on this message on the THIS port issue the following command: `$JASC,PSAT,BLV,1<CR><LF>`
Additional Information	.
Related Commands and Messages	PSAT, BLV message

Topic Last Updated: v1.07 / Octoter 13, 2016

JASC,PSAT,FVI Command

Command Type	Local Differential and RTK
Description	Contains much more special information
Command Format	`$JASC,PSAT,FVI,r[,OTHER]<CR><LF>` where: • **'r' = message rate 0,1,2,5,10,20 (0 = Off, 1 = On at 1Hz)** • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology. .
Receiver Response	`$>`
Example	To turn on this message on the THIS port issue the following command: `$JASC,PSAT,FVI,1<CR><LF>`
Additional Information	.
Related Commands and Messages	PSAT, FVI message

Topic Last Updated: v1.07 / Octoter 13, 2016

JASC,PSAT,RTKPROG Command

Command Type	Local Differential and RTK
Description	Configure the receiver to output RTK fix progress
Command Format	`$JASC,PSAT,RTKPROG,r[,OTHER]<CR><LF>` where: • 'r' = message rate (0 = Off, 1 = On at 1Hz) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology. You can also perform a one-time query of the message information by issuing the JQUERY,RTKPROG command.
Receiver Response	`$>`
Example	To turn on this message on the THIS port issue the following command: `$JASC,PSAT,RTKPROG,1<CR><LF>`
Additional Information	Issuing the JSAVE command after setting JASC,PSAT,RTKPROG to 1 (message on at 1Hz) does not save this setting. You must enable JASC,PSAT,RTKPROG (set it to 1) each time you power on the receiver. See also PSAT,RTKPROG message.

Topic Last Updated: v1.04 / May 29, 2012

JASC,PSAT,RTKSTAT Command

Command Type	Local Differential and RTK
Description	Configure the receiver to output the most relevant parameters affecting RTK
Command Format	`$JASC,PSAT,RTKSTAT,r[,OTHER]<CR><LF>` where: • 'r' = message rate (0 = Off, 1 = On at 1Hz) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology. You can also perform a one-time query of the message information by issuing the JQUERY,RTKSTAT command.
Receiver Response	`$>`
Example	To turn on this message on the THIS port issue the following command: `$JASC,PSAT,RTKSTAT,1<CR><LF>`
Additional Information	Issuing the JSAVE command after setting JASC,PSAT,RTKSTAT to 1 (message on at 1Hz) does not save this setting. You must enable JASC,PSAT,RTKSTAT (set it to 1) each time you power on the receiver.
Related Commands and Messages	JQUERY,RTKSTAT command PSAT,RTKSTAT message

Topic Last Updated: v1.05 / January 18, 2013

JASC,PSAT,VCT Command

Command Type	Local Differential and RTK
Description	
Command Format	`$JASC,PSAT,VCT,r[,OTHER]<CR><LF>` where: • **'r' = message rate 0,1,2,5,10,20 (0 = Off, 1 = On at 1Hz)** • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology. .
Receiver Response	`$>`
Example	To turn on this message on the THIS port issue the following command: `$JASC,PSAT,VCT,1<CR><LF>`
Additional Information	.
Related Commands and Messages	PSAT, VCT message

Topic Last Updated: v1.07 / Octoter 13, 2016

JASC,PTSS1 Command

Command Type

Vector

Description

Configure the receiver to output heave, pitch, and roll in the commonly used TSS1 message format

Command Format

`$JASC,PTSS1,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate (in Hz) of 0 (off), 0.25, 0.5, 1, 2, 4, 5, 10, or 20 (if subscribed)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Receiver Response

`:XXAAAASMHHHHQMRRRRSMPPPP*CC<CR><LF>`

where:

Message Component	Description
XX	Horizontal acceleration
AAAA	Vertical acceleration
HHHH	Heave, in centimeters
S	S = space character
M	Space if positive; minus if negative
Q	Status flag Value — Description h — Heading aided mode (settling) - The System is receiving heading aiding signals from a gyrocompass but is still awaiting the end of the three minutes settling period after power-on or a change of mode or heave bandwidth. The gyrocompass takes approximately five minutes to settle after it has been powered on. During this time, gyrocompass aiding of the System will not be perfect. The status flag does NOT indicate this condition. F — Full aided mode (settled condition) - The System is receiving and using aiding signals from a gyrocompass and from a GPS receiver or a Doppler log.
M	Space if positive; minus if negative
RRRR	Roll, in units of 0.01 degrees (ex: 1000 = 10°)
S	S = space character
M	Space if positive; minus if negative
PPPP	Pitch, in units of 0.01 degrees (ex: 1000 = 10°)
<CR>	Carriage return

<LF>	Line feed

Additional Information TSS1 message

Topic Last Updated: v1.06 / March 10, 2015

JASC,ROX Command

Command Type	Local Differential and RTK
Description	Set the proprietary ROX messages to on or off to provide corrections to the rover This command only applies to an Eclipse base station receiver when using GPS dual frequency RTK mode. RTK is relative to the reference position (base only).
Command Format	`$JASC,ROX,r[,OTHER]<CR><LF>` where: • 'r' = correction status variable (0 = turn corrections Off, 1 = turn corrections On) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.
Receiver Response	`$>`
Example	To turn on ROX messages on the OTHER port issue the following command: `$JASC,ROX,1,OTHER<CR><LF>`
Additional Information	To query the receiver for the current setting, issue the JSHOW command. To change the broadcast station ID, use JRTK,28.

Topic Last Updated: v1.02 / January 25, 2011

JASC,RTCM Command

Command Type	SBAS
Description	Configure the receiver to output RTCM version 2 DGPS corrections from SBAS or beacon through either receiver serial port. The correction data output is RTCM SC-104, even though SBAS uses a different over-the-air protocol (RTCA).
Command Format	`$JASC,RTCM,r[,OTHER]<CR><LF>` where: • 'r' = message status variable (0 = Off, 1 = On) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.
Receiver Response	`$>`
Example	To output RTCM corrections from SBAS or beacon on THIS port (current port) issue the following command: `$JASC,RTCM,1<CR><LF>`
Additional Information	To verify the current setting is on, issue the JSHOW command. You will see output similar to the following: `$>JSHOW,ASC,RTCM,1.0` If the current setting is off, the JSHOW command will not show any information for this setting.

Topic Last Updated: v1.02 / January 25, 2011

JASC,RTCM3 Command

Command Type	Local Differential and RTK
Description	Set the RTCM version 3 messages to on or off to provide corrections to the rover This command only applies to an Eclipse base station receiver when using GPS dual frequency RTK mode. RTK is relative to the reference position (base only).
Command Format	`$JASC,RTCM3,r[,OTHER]<CR><LF>` where: • 'r' = correction status variable (0 = turn corrections Off, 1 = turn corrections On) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.
Receiver Response	`$>`
Example	To turn on RTCM3 messages on the OTHER port issue the following command: `$JASC,RTCM3,1,OTHER<CR><LF>`
Additional Information	To query the receiver for the current setting, issue the JSHOW command. To change the broadcast station ID, use JRTK,28.

Topic Last Updated: v1.02 / January 25, 2011

JASC,VIRTUAL Command

Command Type	General Operation and Configuration
Description	Configure the receiver to have RTCM data input on one port and output through the other (when using an external correction source) For example, if RTCM is input on Port B, the data will be output through Port A having corrected the receiver position. The receiver acts as a pass-through for the RTCM data. Either port may be configured to accept RTCM data input; this command enables the opposite port to output the RTCM data.
Command Format	`$JASC,VIRTUAL,r[,OTHER]<CR><LF>` where: • 'r' = message status variable (0 = Off, 1 = On) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.
Receiver Response	`$>`
Example	To configure THIS port to output RTCM messages that are being input through the OTHER port issue the following command: `$JASC,VIRTUAL,1`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JATT

JATT Command Overview

The JATT command is used to define or query attitude settings for Vector products.

Command	Description
JATT,COGTAU	Set the course over ground (COG) time constant (0.0 to 3600.0 seconds) or query the current setting
JATT,CSEP	Query to retrieve the current separation between GPS antennas
JATT,EXACT	Enable/disable internal filter reliance on the entered antenna separation or query the current setting
JATT,FLIPBRD	Allow upside down installation
JATT,GYROAID	Turn on gyro aiding or query the current feature status
JATT,HBIAS	Set the heading bias or query the current setting
JATT,HELP	Show the available commands for GPS heading operation and status
JATT,HIGHMP	Set/query the high multipath setting for use in poor GPS environments
JATT,HRTAU	Set the rate of turn time constant or query the current setting
JATT,HTAU	Set the heading time constant or query the current setting
JATT,LEVEL	Turn on level operation or query the current feature status
JATT,MOVEBASE	Set the auto GPS antenna separation or query the current setting
JATT,MSEP	Set (manually) the GPS antenna separation or query the current setting
JATT,NEGTILT	Turn on the negative tilt feature or query the current setting
JATT,NMEAHE	Instruct the Vector on how to preface the HDT and HDR messages
JATT,PBIAS	Set the pitch bias or query the current setting
JATT,PTAU	Set the pitch time constant or query the current setting
JATT,ROLL	Configure the Vector for roll or pitch output
JATT,SEARCH	Force a new RTK heading search
JATT,SPDTAU	Set the speed time constant (0.0 to 3600.0 seconds) or query the current setting
JATT,SUMMARY	Show the current configuration of the Vector
JATT,TILTAID	Turn tilt aiding on/off or query the Vector for the current status of this feature
JATT,TILTCAL	Calibrate the internal tilt sensor of the Vector

Topic Last Updated: v1.07 / Octoter 13, 2016

JATT,COGTAU Command

Note: The JTAU,COG command provides identical functionality but works with Crescent and Eclipse products in addition to Crescent Vector products.

Command Type	Vector
Description	Set the course over ground (COG) time constant (0.0 to 3600.0 seconds) or query the current setting This command allows you to adjust the level of responsiveness of the COG measurement provided in the GPVTG message. The default value is 0.0 seconds of smoothing. Increasing the COG time constant increases the level of COG smoothing. COG is computed using only the primary GPS antenna (when using a multi-antenna system) and its accuracy depends upon the speed of the vessel (noise is proportional to 1/speed). This value is invalid when the vessel is stationary, as tiny movements due to calculation inaccuracies are not representative of a vessel's movement.
Command Format	Set the COG time constant `$JATT,COGTAU,cogtau<CR><LF>` where 'cogtau' is the new COG time constant that falls within the range of 0.0 to 200.0 seconds The setting of this value depends upon the expected dynamics of the Crescent. If the Crescent will be in a highly dynamic environment, this value should be set lower because the filtering window would be shorter, resulting in a more responsive measurement. However, if the receiver will be in a largely static environment, this value can be increased to reduce measurement noise. Query the current setting `$JATT,COGTAU<CR><LF>`
Receiver Response	`$>`
Additional Information	You can use the following formula to determine the COG time constant: cogtau (in seconds) = 10 / maximum rate of change of course (in °/s) If you are unsure about the best value for this setting, it is best to be conservative and leave it at the default setting of 0.0 seconds.

Topic Last Updated: v1.06 / March 10, 2015

JATT,CSEP Command

Command Type	Vector
Description	Query the Vector for the current calculated separation between antennas, as solved for by the attitude algorithms
Command Format	`$JATT,CSEP<CR><LF>`
Receiver Response	`$>JATT,X,CSEP` where 'X' is the antenna separation in meters
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JATT,EXACT Command

Command Type	Vector
Description	Enable/disable internal filter reliance on the entered antenna separation or query the current setting
Command Format	Enable/disable internal filter reliance **To enable internal filter reliance:** `$JATT,EXACT,YES<CR><LF>` **To disable internal filter reliance:** `$JATT,EXACT,NO<CR><LF>` Query the current setting `$JATT,EXACT<CR><LF>`
Receiver Response	`$>`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JATT,FLIPBRD Command

Command Type	Vector
Description	Turn the flip feature on/off or query the current feature status Allow the Vector OEM board to be installed upside down. You should use this command only with the Vector Sensor and the Vector OEM board because flipping the OEM board does not affect the antenna array that needs to remain facing upwards. When using this command, the board needs to be flipped about roll so the front still faces the front of the vessel.
Command Format	Turn the flip feature on/off To turn the flip feature on: `$JATT,FLIPBRD,YES<CR><LF>` To turn the flip feature off (return to default mode - right side up): `$JATT,FLIPBRD,NO<CR><LF>` Query current the current setting `$JATT,FLIPBRD<CR><LF>`
Receiver Response	`$>`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JATT,GYROAID Command

Command Type	Vector
Description	Turn gyro aiding on or off or query the current setting The Vector's internal gyro—enabled by default when shipped—offers two benefits. • It shortens reacquisition times when a GPS heading is lost because of obstruction of satellite signals. It does this by reducing the search volume required for solution of the RTK. • It provides an accurate substitute heading for a short period (depending on the roll and pitch of the vessel) ideally seeing the system through to reacquisition. For these two benefits, Hemisphere GNSS highly recommend leaving gyro aiding on. Exceeding rates of 90°/sec is not recommended because the gyro cannot measure rates beyond this point. This is a new recommendation since Hemisphere GNSS now uses gyro measurements to obtain a heading rate measurement.
Command Format	Turn gyro aiding on/off To turn gyro aiding on: `$JATT,GYROAID,YES<CR><LF>` To turn gyro aiding off: `$JATT,GYROAID,NO<CR><LF>` Query the current setting `$JATT,GYROAID<CR><LF>`
Receiver Response	`$>`

Additional Information

Every time you power up the Vector the gyro goes through a warm-up procedure and calibrates itself. You cannot save the resulting calibration, so the self-calibration takes place every time the Vector is power cycled.

This self-calibration procedure takes several minutes and is the equivalent of the following manual calibration procedure.

With the Vector unit installed:

1. Apply power and wait several minutes until it has acquired a GPS signal and is computing heading.
2. Ensure gyroaiding is on by issuing the following command:
   `$JATT,GYROAID<CR><LF>`
3. Slowly spin the unit for one minute at no more than 15°/sec.
4. Keep the unit stationary for four minutes. Both the manual and the self-calibration procedures calibrate the Crescent Vector's gyro to the same effect.

Topic Last Updated: v1.06 / March 10, 2015

JATT,HBIAS Command

Command Type	Vector
Description	Set the heading output from the Vector to calibrate the true heading of the antenna array to reflect the true heading of the vessel or query the current setting
Command Format	Set the heading output `$JATT,HBIAS,x<CR><LF>` where 'x' is a bias that will be added to the Vector's heading in degrees. The acceptable range for the heading bias is -180.0° to 180.0°. The default value of this feature is 0.0°. Query the current setting (current compensation angle) `$JATT,HBIAS<CR><LF>`
Receiver Response	`$>`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JATT,HELP Command

Command Type	Vector
Description	Show the available commands for GPS heading operation and status
Command Format	`$JATT,HELP<CR><LF>`
Receiver Response	`$>JATT,HELP,CSEP,MSEP,EXACT,LEVEL,HTAU,HRTAU,HBIASPBIAS,NEGTILT,ROLL,TILTAID, TILTCAL,MAGAID,MAGCAL,MAGCLR,GYROAID,COGTAU,SPDTAU,SEARCH,SUMMARY`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JATT,HIGHMP Command

Command Type	Vector
Description	Enable/disable the high multipath setting for use in poor GPS environments or query the current setting Enabling HIGHMP mode may result in longer heading acquisition times in high multipath environments. In HIGHMP mode, the Vector will not output heading until it has good confidence in the result. In very poor environments, this may take a few minutes or more; in normal environments, there is only a slight increase in heading acquisition time.
Command Format	Set the high multipath setting **To enable the high multipath setting:** `$JATT,HIGHMP,YES<CR><LF>` **To disable the high multipath setting:** `$JATT,HIGHMP,NO<CR><LF>` Query the current setting `$JATT,HIGHMP<CR><LF>`
Receiver Response	`$>`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JATT,HRTAU Command

Command Type	Vector
Description	Set the rate of turn (ROT) time constant to adjust the level of responsiveness of the ROT measurement provided in the GPROT message or query the current setting The default value of this constant is 2.0 seconds of smoothing. Increasing the time constant increases the level of ROT smoothing.
Command Format	Set the heading rate time constant `$JATT,HRTAU,hrtau<CR><LF>` where 'hrtau' is the new time constant that falls within the range of 0.0 to seconds The setting of this value depends upon the expected dynamics of the vessel. For example, if the vessel is very large and cannot turn quickly, increasing this time is reasonable. The resulting heading would have reduced 'noise', resulting in consistent values with time. However, artificially increasing this value such that it does not agree with a more dynamic vessel could create a lag in the ROT measurement with higher rates of turn. Query the current setting `$JATT,HRTAU<CR><LF>`
Receiver Response	`$>`
Additional Information	You can use the following formula to determine the level of smoothing: hrtau (in seconds) = 10 / maximum rate of the rate of turn (in °/s2) **Note:** If you are unsure about the best value for the setting, leave it at the default setting of 2.0 seconds.

Topic Last Updated: v1.06 / March 10, 2015

JATT,HTAU Command

Command Type	Vector
Description	Set the heading time constant to adjust the level of responsiveness of the true heading measurement provided in the GPHDT message or query the current setting. For OEM boards the default value of this constant is 0.5 seconds of smoothing (regardless of whether the gyro is enabled or disabled). For finished products that implement an OEM board the default value may be different—check your product's documentation for this value. Although the gyro is enabled by default, you can disable it. Increasing the heading time constant increases the level of heading smoothing and increases lag only if the gyro is disabled.
Command Format	Set the heading time constant `$JATT,HTAU,htau<CR><LF>` where 'htau' is the new time constant that falls within the range of 0.0 to seconds The setting of this value depends upon the expected dynamics of the vessel. If the vessel is very large and cannot turn quickly, increasing this time is reasonable. The resulting heading would have reduced 'noise' resulting in consistent values with time. However, artificially increasing this value such that it does not agree with a more dynamic vessel could create a lag in the heading measurement with higher rates of turn. Query the current setting `$JATT,HTAU<CR><LF>`
Receiver Response	`$>`
Additional Information	You can use the following formula to determine level of heading smoothing required when the gyro is in use: Gyro on htau (in seconds) = 40 / maximum rate of turn (in °/s) Gyro off htau (in seconds) = 10 / maximum rate of turn (in °/s) If you are unsure about the best value for the setting, leave it at the default setting of 2.0 seconds when the gyro is on and at 0.5 seconds when the gyro is off.

Topic Last Updated: v1.06 / March 10, 2015

JATT,LEVEL Command

Command Type	Vector
Description	Turn level operation on or off or query the current setting If the Vector will be operated within ±10° of level, you may use this mode of operation for increased robustness and faster acquisition times of the heading solution.
Command Format	Turn level operation on/off To turn level operation on: `$JATT,LEVEL,YES<CR><LF>` To turn level operation off: `$JATT,LEVEL,NO<CR><LF>` Query the current setting `$JATT,LEVEL<CR><LF>`
Receiver Response	`$>`
Additional Information	

Topic Last Updated: v1.05 / January 18, 2013

JATT,MOVEBASE Command

Command Type	Vector
Description	Set the auto GPS antenna separation or query the current setting If the operation is turned on ,you do not need to set the GPS antenna separation manually .But only multi-frequency boards are supported.
Command Format	Turn level operation on/off To turn movebase operation on: `$JATT,MOVEBASE,YES<CR><LF>` To turn movebase operation off: `$JATT,MOVEBASE,NO<CR><LF>` Query the current setting `$JATT,MOVEBASE<CR><LF>`
Receiver Response	`$>`
Additional Information	

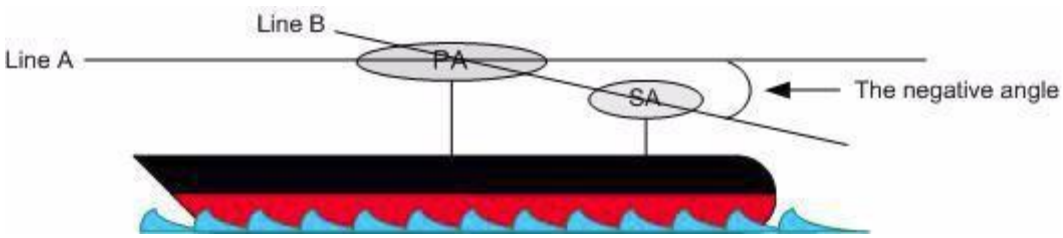
Topic Last Updated: v1.07 / Octoter 13, 2016

JATT,MSEP Command

Command Type	Vector
Description	Manually enter a custom separation between antennas (must be accurate to within 1 to 2 cm) or query the current setting
Command Format	Set the antenna separation Using the new center-to-center measurement, issue the following command: `$JATT,MSEP,sep<CR><LF>` where 'sep' is the measured antenna separation entered in meters Query the current setting `$JATT,MSEP<CR><LF>`
Receiver Response	`$>`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JATT,NEGTILT Command

Command Type	Vector
Description	Turn the negative tilt feature on or off or query the current setting. When the secondary GPS antenna (SA) is below the primary GPS antenna (PA), there is an angle formed between a horizontal line through the center of the primary antenna (Line A in the diagram below) and an intersecting line through the center of the primary and secondary antennas (Line B). This angle is considered to be negative.  Depending on the convention for positive and negative pitch/roll, you want to change the sign (either positive or negative) of the pitch/roll.
Command Format	Turn negative tilt feature on/off To change the sign of the pitch/roll measurement: `$JATT,NEGTILT,YES<CR><LF>` To return the sign of the pitch/roll measurement to its original value: `$JATT,NEGTILT,NO<CR><LF>` Query the current setting `$JATT,NEGTILT<CR><LF>`
Receiver Response	`$>`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JATT,NMEAHE Command

Command Type	Vector
Description	Instruct the Vector to preface the following messages with GP or HE. • HDG • HDM • HDT • ROT
Command Format	`$JATT,NMEAHE,x<CR><LF>` where 'x' is either 1 for HE or 0 for GP To preface specific messages with GP `$JATT,NMEAHE,0<CR><LF>` To preface specific messages with HE `$JATT,NMEAHE,1<CR><LF>`
Receiver Response	`$>JATT,NMEAHE,OK`
Additional Information	The HDM message is for a magnetic compass. The message will be **HC**HDM when requesting with `$JATT,NMEAHE,1` specified.

Topic Last Updated: v1.06 / March 10, 2015

JATT,PBIAS Command

Command Type	Vector
Description	Set the pitch/roll output from the Vector to calibrate the measurement if the antenna array is not installed in a horizontal plane or query the currentsetting
Command Format	Set the pitch/roll output `$JATT,PBIAS,x<CR><LF>` where 'x' is a bias that will be added to the Vector's pitch/roll measure, in degrees The acceptable range for the pitch bias is -15.0° to 15.0°. The default value is 0.0°. Query the current setting `$JATT,PBIAS<CR><LF>`
Receiver Response	`$>`
Additional Information	**Note:** The pitch/roll bias is added after the negation of the pitch/roll measurement (if invoked with the JATT,NEGTILT command). Use PBIAS to describe any angular differences between the level of the two GPS antennas. Pitch is the default, but if the antennas are mounted in the roll direction, you can still enter the roll bias in PBIAS (make sure JATT,ROLL,YES isset).

Topic Last Updated: v1.06 / March 10, 2015

JATT,PTAU Command

Command Type	Vector
Description	Set the level of responsiveness of the pitch measurement provided in the PSAT,HPR message or query the current setting. For OEM boards the default value of this constant is 0.5 seconds of smoothing (regardless of whether the gyro is enabled or disabled). For finished products that implement an OEM board the default value may be different—check your product's documentation for this value. Increasing the pitch time constant increases the level of pitch smoothing and increases lag.
Command Format	Set the pitch time constant `$JATT,PTAU,ptau<CR><LF>` where 'ptau' is the new time constant that falls within the range of 0.0 to 3600.0 seconds. The setting of this value depends upon the expected dynamics of the vessel. For instance, if the vessel is very large and cannot pitch quickly, increasing this time is reasonable. The resulting pitch would have reduced 'noise', resulting in consistent values with time. However, artificially increasing this value such that it does not agree with a more dynamic vessel could create a lag in the pitch measurement. Query the current setting `$JATT,PTAU<CR><LF>` **Note:** If you are unsure about the best value for the setting, leave it at the default setting of 0.5 seconds.
Receiver Response	`$>`
Additional Information	You can use the following formula to determine the level of pitch smoothing required: ptau (in seconds) = 10 / maximum rate of pitch (in °/s)

Topic Last Updated: v1.06 / March 10, 2015

JATT,ROLL Command

Command Type

Vector

Description

Configure the Vector for roll or pitch GPS antenna orientation.

Command Format

Configure the Vector for pitch or roll GPS antenna orientation

To configure the Vector for roll GPS antenna orientation (the Antenna Array must be installed perpendicular to the vessel's axis):

```
$JATT,ROLL,YES<CR><LF>
```

To configure the Vector for pitch GPS antenna orientation (default):

```
$JATT,ROLL,NO<CR><LF>
```

Query the current setting

```
$JATT,ROLL<CR><LF>
```

Receiver Response

```
$>
```

Additional Information

Topic Last Updated: v1.06 / March 10, 2015

JATT,SEARCH Command

Command Type	Vector
Description	Force the Vector to reject the current GPS heading solution and begin a new search.
Command Format	`$JATT,SEARCH<CR><LF>`
Receiver Response	$>
Additional Information	The SEARCH function will not work if you have enabled the gyroaid feature (using the GYROAID command). In this case you must cycle power to the receiver to have a new GPS solution computed.

Topic Last Updated: v1.06 / March 10, 2015

JATT,SPDTAU Command

Note: **The JTAU,SPEED command provides identical functionality but works with Crescent and Eclipse products in addition to Crescent Vector products.**

Command Type	Vector
Description	Set the speed time constant (0.0 to 3600.0 seconds) or query the current setting. This command allows you to adjust the level of responsiveness of the speed measurement provided in the GPVTG message. The default value is 0.0 seconds of smoothing. Increasing the speed time constant increases the level of speed measurement smoothing.
Command Format	Set the speed time constant `$JATT,SPDTAU,spdtau<CR><LF>` where 'spdtau' is the new time constant that falls within the range of 0.0 to 200.1 seconds The setting of this value depends upon the expected dynamics of the receiver. If the receiver will be in a highly dynamic environment, you should set this to a lower value, since the filtering window will be shorter, resulting in a more responsive measurement. However, if the receiver will be in a largely static environment, you can increase this value to reduce measurement noise. Query the current setting `$JATT,SPDTAU<CR><LF>`
Receiver Response	`$>`
Additional Information	You can use the following formula to determine the COG time constant (Hemisphere GNSS recommends testing how the revised value works in practice): spdtau (in seconds) = 10 / maximum acceleration (in m/s^2) If you are unsure about the best value for this setting, it is best to be conservative and leave it at the default setting of 0.00 seconds.

Topic Last Updated: v1.06 / March 10, 2015

JATT,SUMMARY Command

Command Type Vector

Description Display a summary of the current Vector settings

Command Format

```
$JATT,SUMMARY<CR><LF>
```

Receiver Response

```
$>JATT,SUMMARY,htau,hrtau,ptau,cogtau,spdtau,hbias,pbias,hexflag<CR><LF>
```

where:

Component	Description
htau	Current heading time constant, in seconds
hrtau	Current heading rate time constant, in seconds
ptau	Current pitch time constant, in seconds
cogtau	Current course over ground time constant, in seconds
spdtau	Current speed time constant, in seconds
hbias	Current heading bias, in degrees
pbias	Current pitch/roll bias, in degrees
hexflag	Hex code that summarizes the heading feature status Flag / 'On' Value / 'Off' Value Gyro aiding / 02 / 0 Negative tilt / 01 / 0 Roll / 08 / 0 Tilt aiding / 02 / 0 Level / 01 / 0

The 'hexflag' field is two separate hex flags:

- 'GN' - Value is determined by computing the sum of the gyro aiding and negative tilt values, depending on whether they are on or off:

- If the feature is on, their value is included in the sum
- If the feature is off, it has a value of zero when computing the sum
- 'RMTL' - Value is determined in much the same way but by adding the values of roll, tilt aiding, and level operation.

For example, if gyro aiding, roll, and tilt aiding features were each on, the values of 'GN' and 'RMTL' would be:

- **'GN' = hex (02 + 0) = hex (02) = 2**
- **'RMTL' = hex (08 + 02) = hex (10) = A**
- ‘GN-RMTL’ = 2A

The following tables summarize the possible feature configurations for the first 'GN' character and the second 'RMTL' character.

JATT,SUMMARY 1st GN Character Configurations		
GN Value	Gyro Value	Negative Tilt
0	Off	Off
1	Off	On
2	On	Off
3	On	On

JATT,SUMMARY 2nd RMTL Character Configurations			
RMTL Value	Roll	Tilt Aiding	Level
0	Off	Off	Off
1	Off	Off	On
2	Off	On	Off
3	Off	On	On
8	On	Off	Off
9	On	Off	On
A	On	On	Off
B	On	On	On

Example `$>JATT,SUMMARY,TAU:H=0.50,HR=2.00,COG=0.00,SPD=0.00,BIAS:H=0.00,P=0.00, FLAG_HEX:HF-RMTL=01`

Additional Information

Topic Last Updated: v1.06 / March 10, 2015

JATT,TILTAID Command

Command Type	Vector
Description	Turn tilt aiding on or off or query the current setting. The Vector's internal tilt sensors (accelerometers) may be enabled by default (see your specific product manuals for further information). The sensors act to reduce the RTK search volume, which improves heading startup and reacquisition times. This improves the reliability and accuracy of selecting the correct heading solution by eliminating other possible, erroneous solutions.
Command Format	Turn tilt aiding on/off Turn tilt aiding on: `$JATT,TILTAID,YES<CR><LF>` Turn tilt aiding off: `$JATT,TILTAID,NO<CR><LF>` Query the current setting `$JATT,TILTAID<CR><LF>`
Receiver Response	Response to issuing command to turn tilt aiding on/off `$>` Response to querying the current setting If setting is currently ON the response is: `$>JATT,TILTAID,ON` If setting is currently OFF the response is: `$>JATT,TILTAID,OFF`
Additional Information	Tilt aiding is required to increase the antenna separation of the Vector OEM beyond the default 0.5 m length.

Topic Last Updated: v1.06 / March 10, 2015

JATT,TILTCAL Command

Command Type	Vector
Description	Calibrate the internal tilt sensors of the Vector. Calibration takes approximately two seconds and is automatically saved to memory for subsequent power cycles. You can calibrate the tilt sensor of the Vector in the field but the Vector enclosure must be horizontal when you calibrate.
Command Format	`$JATT,TILTCAL<CR><LF>`
Receiver Response	`$>`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JBAUD Command

Command Type	General Operation and Configuration
Description	Specify the baud rates of the receiver or query the current setting.

Command Format

Specify the baud rates

```
$JBAUD,r[,OTHER][,SAVE]<CR><LF>
```

where:

- **'r' = baud rate (4800, 9600, 19200, 38400, 57600, or 115200)**
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)
- ',SAVE' = optional field, saves the baud rate into flash memory so that if you reset power the receiver will boot at the new baud rate (it may take several seconds to save the baud rate to flash memory)

Query the current setting

```
$JBAUD[,OTHER]<CR><LF>
```

where:

- ',OTHER' = optional field, queries the current port when you send the command without it (and without the brackets) and queries the other port when you send the command with it (without the brackets)

Receiver Response

```
$>JBAUD,R[,OTHER]
```

The response format is the same whether you specify the baud rates or query the current settings.

Example

Issue the following command to set the baud rate to 19200 on the current port:

```
$JBAUD,19200<CR><LF>
```

...the response is then:

```
$>JBAUD,19200
```

Issue the following command to set the baud rate to 9600 on the OTHER port and save it into memory:

```
$JBAUD,9600,OTHER,SAVE<CR><LF>
```

...the response is then:
`$>JBAUD,9600,OTHER`

Additional Information	**Note:** When saving the baud rate wait until you see the SAVE COMPLETE message before powering off the receiver. See the JSAVE command for an example of this output. The status of this command is also output when issuing the JSHOW command.

Topic Last Updated: v1.02 / January 25, 2011

JBIN Command

Command Type General Operation and Configuration

Description Enable the output of the various binary messages—most notably the Bin95 and Bin96 messages—to be requested. The Bin95 and Bin96 messages contain all the information required for post processing.

Command Format

```
$JBIN,msg,r<CR><LF>
```

where:

- 'msg' = binary message you want to output
- 'r' = message rate as shown in the following table

Message Name	MSG	R (Hz)	Description
Bin1	1	20, 10, 2, 1, 0, or .2	GPS position message (position and velocity data)
Bin2	2	1 or 0	GPS DOPs (Dilution of Precision)
Bin3	3	20, 10, 2, 1, 0, or .2	Lat/Lon/Hgt, Covariances, RMS, DOPs and COG, Speed, Heading
Bin16	16		All constellation code and phase observation data
Bin19			GNSS diagnostic information
Bin35	35	1 or 0	BeiDou ephemeris information
Bin36	36	1 or 0	BeiDou code and carrier phase information (all frequencies)
Bin44	44		GALILEO time conversion
Bin45	45		GALILEO ephemeris
Bin62	62	1 or 0	GLONASS almanac information
Bin65	65	1 or 0	GLONASS ephemeris information
Bin66	66	20, 10, 2, 1, or 0	GLONASS L1/L2 code and carrier phase information
Bin69	69	1 or 0	GLONASS L1/L2 diagnostic information
Bin76	76	20, 10, 2, 1, 0, or .2	GPS L1/L2 code and carrier phase information
Bin80	80	1 or 0	SBAS data frame information
Bin89	89	1 or 0	SBAS satellite tracking information
Bin93	93	1 or 0	SBAS ephemeris information
Bin94	94	1 or 0	Ionospheric and UTC conversion parameters
Bin95	95	1 or 0	GPS ephemeris information
Bin96	96	20, 10, 2, 1, or 0	GPS L1 code and carrier phase information
Bin97	97	20, 10, 2, 1, 0, or .2	Processor statistics

Bin98	98	1 or 0	GPS satellite and almanac information
Bin99	99	1 or 0	GPS L1 diagnostic information
Bin100	100	1 or 0	GPS L2 diagnostic information
Bin209	209	1 or 0	SNR and status for all GNSS tracks

Receiver Response `$>`

Example To output the Bin76 message at a rate of 10 Hz, issue the following command:

```
$JBIN,76,10<CR><LF>
```

Additional Information Higher update rates may be available with a subscription on Bin 1, 2, 96, 97 and 99.

Topic Last Updated: v1.07 / February 16, 2017

JBOOT

JBOOT Command

Command Type	[General Operation and Configuration]
Description	**Power down the Eclipse engine and then power it back up. This allows you to reboot the receiver to drop the satellite to which it is currently locked and retune to another satellite without cycling the power of the Eclipse II.**
Command Format	`$JBOOT<CR><LF>`
Receiver Response	If MFA is the current application and you send the $JBOOT command, the response is similar to the following: `$>STARTED,MFA,Ver=1.2Qe` If any application other than MFA is the current application and you send the $JBOOT command, the response is similar to the following: $>
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JBOOT,LBAND Command

Command Type	L-Band
Description	Power down theAtlas portion of the Eclipse engine and then power it back up. This allows you to reboot the receiver to drop the satellite to which it is currently locked and retune to another satellite without cycling the power of the Eclipse II..
Command Format	`$JBOOT,LBAND<CR><LF>`
Receiver Response	`$>`
Additional Information	JFREQ

Topic Last Updated: v1.07 / February 16, 2017

JCONN Command

Command Type

[General Operation and Configuration](#)

Description

Create a virtual circuit between two ports to enable communication through the receiver to the device on the opposite port.

Command Format

To connect two ports virtually:

```
$JCONN,P1,P2<CR><LF>
```

where P1 and P2 are a pair of the following: A,B,C,D or PortA,PortB,PortC,PortD

Examples

```
$JCONN,A,B<CR><LF>

$JCONN,PortA,PortB<CR><LF>
```

To disconnect virtual connection:

```
$JCONN,X<CR><LF>
```

Receiver Response

```
$>
```

Additional Information

Caution: Hemisphere GNSS receivers with menus, such as an R Series, use JCONN within the menu application. Any settings you make with JCONN on these products may disable the menu functions until power is cycled.

Topic Last Updated: v1.06 / March 10, 2015

JDIFF

JDIFF Command

Command Type	General Operation and Configuration
Description	Specify or query the differential source of the receiver. Forces the system to use “diff” as the source (see table in Command Format section below).
Command Format	Specify the differential mode `$JDIFF,diff[,SAVE]<CR><LF>`

where:

- 'diff' (differential source) may be one of the following:

DIFF	Description
OTHER	Instruct the receiver to use external corrections input through the opposite port that is communicating
THIS	Instruct the receiver to use external corrections input through the same port that is communicating
PORTA or PORTB or PORTC or PORTD	Instruct the receiver to: • Use external corrections input through the specified port. • Allow RTCM2 (DGPS) inputs to receiver.
BEACON	Instruct the receiver to use RTCM corrections entering Port C at a fixed rate of 9600 baud. This input does not have to be from a beacon receiver, such as SBX. However, this is a common source of corrections.
WAAS	Instruct the receiver to use SBAS. This is also the response when running the local dif application as the base.
RTK	Response when running the local dif or rover RTK application for the rover.
LBAND	Instruct the receiver to turn on theAtlas module and useAtlas. Setting diff to anything other thanAtlas turns off theAtlas module.
X	Instruct the receiver to use e-Dif

	mode
NONE	Instruct the receiver to operate in autonomous mode. This turns off the use of SBAS,Atlas, and RTCM2 (DGPS); however, RTK is still allowed.

- ,SAVE' = optional field, saves the differential source into flash memory so that if you reset power the receiver will boot with the new differential source (it may take several seconds to save the differential source to flash memory).
- Using $JDIFF with SBAS, RTCM2, or Atlas assigns the priority in the MFA. For example, RTCM2 is a higher priority if the assigned diff port is PORTA. See MFA for more information.

Query the current DIFF setting

```
$JDIFF<CR><LF>
```

Receiver Response

Receiver response when specifying the differential source

$>

Receiver response when querying the differential source

```
$>JDIFF,SOURCE,TYPE
```

where:

- 'SOURCE' is the port/source as issued with the JDIFF command
- 'TYPE' is the differential type actually being used

'AUTO' is the response when queried in e-Dif

Example

Issue the following command to query the receiver:

```
$JDIFF<CR><LF>
```

...and if the differential source is WAAS, the response is:

```
$>JDIFF,WAAS
```

Additional Information

The status of this command is also output in the JSHOW message.

Topic Last Updated: v1.07/ February 16, 2017

JDIFF,AVAILABLE Command

Command Type	General Operation and Configuration
Description	Query the receiver for the differential types currently being received
Command Format	`$JDIFF,AVAILABLE<CR><LF>`
Receiver Response	`$>JDIFFX,AVAILABLE,x[,x][,x]...[,x]` where 'x' is the differential type(s)
Example	
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JDIFFX,EXCLUDE Command

Command Type	General Operation and Configuration
Description	Specify the differential sources to be excluded from operating in a multi-differential application or query the receiver for excluded differential sources
Command Format	Specify the differential sources to be excluded `$JDIFFX,EXCLUDE[,SBAS][,RTCM2][,EDIF][,DFX][,CMR] [,RTCM3][,ROX]<CR><LF>` Query the current setting `$JDIFFX,EXCLUDE<CR><LF>`
Receiver Response	Response to issuing command to exclude differential sources $> Response to querying the current setting `$JDIFFX,EXCLUDE[,SOURCE1][,SOURCE2]...[,SOURCEn]<CR><LF>` where SOURCE1 through SOURCEn represent each excluded source
Example	Issue the following command to exclude RTCM3: `$JDIFFX,EXCLUDE,RTCM3<CR><LF>` If you then issue `$JDIFFX,EXCLUDE<CR><LF>` to query the current setting the response is (if RTCM3 is the only excluded source): `$>JDIFFX,EXCLUDE,RTCM3<CR><LF>`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

JDIFFX,GNSSOUT Command

Command Type

General Operation and Configuration

Description

Specify the GNSS systems to be output in the differential or query the current setting

Command Format

Specify the GNSS systems to be output in the differential

$JDIFFX,GNSSOUT,gnss,x<CR><LF>

where:

- 'gnss' = GNSS system to be output in the differential (GPS, GLONASS, BEIDOU, GALILEO)
- 'x' = NO (do not output specified GNSS system in the differential) or YES (output specified GNSS system in the differential)

Query the current setting

Query what GNSS systems are output in the differential

$JDIFFX,GNSSOUT<CR><LF

Query if a specific GNSS system is output in the differential

$JDIFFX,GNSSOUT,gnss<CR><LF

where 'gnss' is the GNSS system

Receiver Response

Receiver response when specifying the GNSS systems to be output in the differential

$>

Receiver response when querying the current setting

See Example section below

Example

Specify that GPS is output in correction formats

Command: $JDIFFX,GNSSOUT,GPS,YES<CR><LF>

Response: $>

-

Query what GNSS systems are output in the differential

Command: $JDIFFX,GNSSOUT<CR><LF>

Response if just GPS: $>JDIFFX,GNSSOUT,GPS

Response if all GPS and GLONASS:
$>JDIFFX,GNSSOUT,GPS,GLONASS

Query if a specific GNSS system is output in the differential (example uses GLONASS)

Command: $JDIFFX,GNSSOUT,GLONASS<CR><LF>

Response if GLONASS is output:
$>JDIFFX,GNSSOUT,GLONASS,YES

Response if GLONASS is not output:

```
$>JDIFFX,GNSSOUT,GLONASS,NO
```

Additional Information

Topic Last Updated: v1.07 / February 16, 2017

JDIFFX,INCLUDE Command

Command Type

General Operation and Configuration

Description

Specify the differential sources to be allowed to operate in a multi-differential application or query the receiver for included differential sources

Command Format

Specify the differential sources to be included

`$JDIFFX,INCLUDE[,SBAS][,RTCM2][,EDIF][,DFX][,CMR][,RTCM3][,ROX ][,ATLAS]<CR><LF>`

Query the current setting

`$JDIFFX,INCLUDE<CR><LF>`

Receiver Response

Response to issuing command to include differential sources

$>

Response to querying the current setting

`$JDIFFX,INCLUDE[,SOURCE1][,SOURCE2]...[,SOURCEn]<CR><LF>`

where SOURCE1 through SOURCEn represent each included source

Example

Issue the following command to include CMR:

`$JDIFFX,INCLUDE,CMR<CR><LF>`

If you then issue `$JDIFFX,INCLUDE<CR><LF>` to query the current setting the response may be (showing all included sources including CMR):

`$>JDIFFX,INCLUDE,SBAS,RTCM2,EDIF,DFX,CMR,RTCM3,ROX`

Additional Information

For example, if an Eclipse II receiver with SBAS,Atlas, and RTK-base in the same application (multi-diff) has no activeAtlas subscription:

1. The receiver triesAtlas high precision services and when it is not found, falls back toAtlas DGPS service.
2. The receiver triesAtlas DGPS service and when it is not found, falls back to WAAS.
3. No warnings when subscription has expired – user expects a certain level of accuracy withAtlas services, not SBAS level accuracy.

If you do not actively watch theAtlas service end date, you could potentially use SBAS without knowing it. This command limits the differential sources to ensure a certain level of accuracy is retained.

Topic Last Updated: v1.07 / February 16, 2017

JDIFFX,SOURCE Command

Command Type	General Operation and Configuration
Description	Query the receiver for the differential source
Command Format	`$JDIFFX,SOURCE<CR><LF>`
Receiver Response	`$>JDIFFX,source` where 'source' is the differential source
Example	Response ifAtlas is thedifferential source `$>JDIFFX,SOURCE,LBAND` Response if RTK is the differential source through Port B `$>JDIFFX,SOURCE,PORTB`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JDIFFX,TYPE Command

Command Type	General Operation and Configuration
Description	Query the receiver for the differential type
Command Format	`$JDIFFX,TYPE<CR><LF>`
Receiver Response	`$>JDIFFX,TYPE,type` where 'type' is one of the following differential types: • NONE (no differential corrections) • CMR • DFX • EDIF • ROX • RTCM2 • RTCM3 • SBAS
Example	Response if SBAS is the differential type `$>JDIFFX,TYPE,SBAS` Response if RTK (ROX) is the differential type `$>JDIFFX,TYPE,ROX`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JEPHOUT,PERIODSEC Command

Command Type [General Operation and Configuration]

Description to allow ephemeris messages (95, 65, 35) to go out a rate other than when they change. This also does the same rate for the ionoutc message 94. This is a global message and applies to all ephemeris messages on all ports..

Command Format

Enable/disable the command

To enable this command

`$JEPHOUT,1<CR><LF>`

To disable this command:

`$JEPHOUT,0<CR><LF>`

Query the current setting

`$JEPHOUT<CR><LF>`

Receiver Response

Response to issuing command to enable/disable command

$>

Response to querying the current setting

If setting is currently enabled the response is:

`$>JEPHOUT,1`

If setting is currently disabled the response is:

`$>JEPHOUT,0`

Additional Information ..

Topic Last Updated: v1.07 / Octoter 13, 2016

JETHERNET

JETHERNET Command Overview

The JETHERNET command is used to configure Ethernet settings on Ethernet-capable boards.

Command	Description
JETHERNET	Query current Ethernet configuration state
JETHERNET,MODE	Enable/Disable Ethernet
JETHERNET, PORTI	Enable/Disable PORTI virtual serial port

Topic Last Updated: v1.07 / February 16, 2017

JETHERNET,MODE

Command Type General Operation and Configuration

Description On receivers with Ethernet support, this command allows configuring how the receiver connects to a network on the Ethernet interface.

Command Format

```
$JETHERNET,MODE,OFF<CR><LF>
$JETHERNET,MODE,DHCP<CR><LF>
$JETHERNET,MODE,STATIC,IP,SUBNET[,GATEWAY[,DNS]]<CR><LF>
Where IP, SUBNET, GATEWAY, and DNS are the ip address, subnet
mask, gateway ip, and dns server ip respectively, in the standard
decimal notation.
```

Receiver Response

```
$>JETHERNET,MODE,...<CR><LF>
```

Example To disable Ethernet support, one would use the command

```
$JETHERNET,MODE,OFF<CR><LF>
To enable Ethernet support in DHCP (automatic IP address
assignment by the network) mode, use the following command.
$JETHERNET,MODE,DHCP<CR><LF>
To enable Ethernet support with a fixed IP address of
192.168.1.5, one could use the following command.
$JETHERNET,MODE,STATIC,192.168.1.5,255.255.255.0<CR><LF>
```

Additional Information

Topic Last Updated v.1.07 / : February 16, 2017

JETHERNET,PORTI

Command Type	General Operation and Configuration
Description	This command configures the virtual serial port 'PORTI', which may be accessible via the Ethernet interface. By default PORTI is disabled, but may be enabled on a specified TCP port using this command. Note that PORTI provides full access just as a local serial port would, without any authentication, so should only be enabled on a trusted network.
Command Format	`$JETHERNET,PORTI,OFF<CR><LF>` `$JETHERNET,PORTI,TCPPORT<CR><LF>` `Where TCPPORT is a decimal number from 1 to 65535 representing the TCP port to listen for incoming connections on.`
Receiver Response	`$>JETHERNET,PORTI,...<CR><LF>` Where the response reflects the current configuration.
Example	To disable the PORTI virtual serial port, one may use the command: `$>JETHERNET,PORTI,OFF<CR><LF>` **To enable PORTI listening on TCP port 5000, one may use the following command:** `$>JETHERNET,PORTI,5000<CR><LF>`
Additional Information	

Topic Last Updated: v1.07 / February 14. 2017

JFLASH

JFLASH Command Overview

The JFLASH command is used to perform file operations via a USB flash drive on Eclipse and Eclipse II based receivers.

Command	Description
JFLASH,DIR	Display the files on a USB flash drive
JFLASH,FILE,CLOSE	Close an open file on a USB flash drive
JFLASH,FILE,NAME	Open a specific file, append to a specific file, or display the file name of the open file on a USB flash drive
JFLASH,FILE,OPEN	Create and open a file with an automatically generated file name on a USB flash drive
JFLASH,FREESPACE	Display the free space in kilobytes (KB) on a USB flash drive
JFLASH,NOTIFY,CONNECT	Enable/disable the automatic response when a USB flash drive is inserted or removed
JFLASH,QUERYCONNECT	Manually verify if a USB flash drive is connected or disconnected

Topic Last Updated: v1.02 / January 25, 2011

JFLASH,DIR Command

Command Type	General Operation and Configuration
Description	Display the files on a USB flash drive You can only display files at the root level of the flash drive (you cannot navigate into subdirectories).
Command Format	`$JFLASH,DIR<CR><LF>`
Receiver Response	`$>JFLASH,file1` `$>JFLASH,file2` `$>JFLASH,file3` `...` `$>JFLASH,filen` One line appears for each file at the root level of the flash drive.
Example	If you issue the `$JFLASH,DIR` command and the root level of the flash drive contains the following files: hemi_1.bin, hemi_2.bin, hemi_3.bin the response is: `$>JFLASH,hemi_1.bin` `$>JFLASH,hemi_2.bin` `$>JFLASH,hemi_3.bin`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JFLASH,FILE,CLOSE Command

Command Type	General Operation and Configuration
Description	Close an open file on a USB flash drive Closing a file does not turn off the messages being written to the flash drive; it just closes the file so you can safely remove the flash drive. **Caution:** Close the file before removing the flash drive. Failure to do so may corrupt the file.
Command Format	`$JFLASH,FILE,CLOSE<CR><LF>`
Receiver Response	`$>JFLASH,CLOSE mass_storage:0:\filename`
Example	If you issue the `$JFLASH,FILE,CLOSE` command and the 'hemi_4.bin' file on the flash drive is currently open, the response is: `$>JFLASH,CLOSE mass_storage:0:\HEMI_4.BIN`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JFLASH,FILE,NAME Command

Command Type [General Operation and Configuration]

Description Open a specific file, append to a specific file, or display the file name of the open file on a USB flash drive

Command Format

Open a specific file (overwrite or append)

```
$JFLASH,FILE,NAME,filename[,APPEND]<CR><LF>
```

where:

- 'filename' is the name of the file and it must be a legal 8.3 file name
- ',APPEND' is an optional field that allows you to append data to the file

Warning: Using this command without the ',Append' option overwrites the existing file without warning.

Display the name of the open file

```
$JFLASH,FILE,NAME<CR><LF>
```

Receiver Response

Response from issuing command to open an existing file or append to an existing file

```
$>JFLASH, OPEN mass_storage:0:\filename
```

Response from issuing command to display the name of the open file

```
$>JFLASH, mass_storage:0:\filename
```

If you attempt to display the name of the open file and no file is actually open the response is:

```
$>JFLASH, NO FILE OPEN
```

Example If you issue the following command to open file hemi_4.bin on a USB flash drive:

```
$JFLASH,FILE,NAME,hemi_4.bin<CR><LF>
```

the response is:

```
$>JFLASH, mass_storage:0:\HEMI_4.BIN
```

Additional Information

Topic Last Updated: v1.02 / January 25, 2011

JFLASH,FILE,OPEN Command

Command Type	General Operation and Configuration
Description	Create and open a file with an automatically generated file name (hemi_1.bin … hemi_99.bin) on a USB flash drive (only 8.3 file format is allowed)
Command Format	`$JFLASH,FILE,OPEN<CR><LF>`
Receiver Response	`$>JFLASH,OPEN mass_storage:0:\filename` where 'filename' is the name of the new file
Example	If you issue the `$JFLASH,FILE,OPEN` command and the root level of the flash drive contains the following files: hemi_1.bin, hemi_2.bin, hemi_3.bin the response is: `$>JFLASH,OPEN mass_storage:0:\HEMI_4.bin`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JFLASH,FREESPACE Command

Command Type	General Operation and Configuration
Description	Display the free space in kilobytes (KB) on a USB flash drive You can use a flash drive larger than 4GB; however, this command will not display a number greater than 4GB.
Command Format	`$JFLASH,FREESPACE<CR><LF>`
Receiver Response	`$>JFLASH,FREESPACE,   numbytes bytes` where 'numbytes' is the number of kilobytes
Example	The following response indicates a USB flash drive with approximately 2GB of free space. `$>JFLASH,FREESPACE,     2001731584 bytes`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JFLASH,NOTIFY,CONNECT Command

Command Type	General Operation and Configuration
Description	Enable/disable the automatic response when a USB flash drive is inserted or removed (if port is not specified the response will be sent to the port that issued the command)
Command Format	`$JFLASH,NOTIFY,CONNECT,r[,PORT]<CR><LF>` where: • 'r' is the message status variable (0 = Off, 1 = On) • ',PORT' is an optional field you use to specify the port to which the response will be sent (if you do not specify a port, the response is sent to the port from which you issued the command)
Receiver Response	Response to issuing command to enable notification `$>` Response to inserting a flash drive if notification is enabled `$>JFLASH,CONNECTED` Response to removing a flash drive if notification is enabled `$>JFLASH,DISCONNECTED`
Additional Information	

JFLASH,QUERYCONNECT Command

Command Type	General Operation and Configuration
Description	Manually verify if a USB flash drive is connected or disconnected
Command Format	`$JFLASH,QUERYCONNECT<CR><LF>`
Receiver Response	Response to verifying the connection status of a flash drive if the flash drive is connected `$>JFLASH,CONNECTED` `$>` Response to verifying the connection status of a flash drive if the flash drive is disconnected `$>JFLASH,DISCONNECTED` `$>`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JFREQ Command

Command Type [L-Band]

Description Tune the Atlas receiver (manually or automatically) or query the receiver for the current setting

Command Format

Tune the Atlas receiver To manually tune the receiver:

`$JFREQ,freq,symb<CR><LF>`

where:

- 'freq' is the frequency in kHz (reply is in MHz)
- 'symb' is the symbol baud rate (1200 or 2400)

Note: When manually tuning the receiver by entering the frequency ('freq') make sure you enter a decimal point before the last digit for any frequencies that are to .5 Hz (see table in Additional Information section below).
Examples:
Correct: `$JFREQ,1557835,1200` (1,557,835 Hz, no decimal required)
Correct: `$JFREQ,1539962.5,600` (1,539,962.5 Hz, decimal required)
Incorrect: `$JFREQ,15399625,600` (1,539,962.5 Hz, decimal required)

To auto-tune the receiver:

`$JFREQ,0<CR><LF>`

Note: You must restart theAtlas receiver (either by cycling power to theAtlas receiver or by issuing the JBOOT,LBAND command) for changes to take effect.

Query the current setting

`$JFREQ<CR><LF>`

Receiver Response

Response to issuing command to tune receiver

$>

Response to querying the current setting

$>JLBEAM,Sent sfreq,Used ufreq,Baud baud,Geolon[,AUTO]

where:

Response Component	Description
sfreq	Frequency to which theAtlas receiver is instructed to tune (in this example, 1557.8550 MHz)
ufreq	Frequency to which theAtlas receiver is tuned
baud	Baud rate of the signals being received
lon	Approximate longitude of the geostationary satellite to which theAtlas receiver is tuned

AUTO	[Optional Field] 'AUTO' appears at the end of the query response only when the L-band receiver is in 'auto-tune' mode.

Example

Manually Tune a Frequency (command and response)

```
$JFREQ,1557835,1200

$>
```

Auto-Tune a Frequency based on Geographic Location (command and response)

```
$JFREQ,0

$>
```

Query a Manually Tuned Receiver (response)

```
$>JLBEAM,Sent 1557.8350,Used 1557.8350,Baud 1200,Geo -101
```

Query an Auto-Tuned Receiver (response)

```
$>JLBEAM,Sent 1557.8550,Used 1557.8550,Baud 1200,Geo -101,AUTO
```

Additional Information

The status of this command is also output when issuing the JSHOW command.

The following table provides frequency information for the Atlas satellites. This information is subject to change. Visit your Atlas service provider's website for up-to-date satellite constellation and broadcast information.

Coverage Area	Frequency	Baud Rate	Satellite Name
North and South America	1545.5300	600	AMERICAS
Asia-Pacific	1539.8525	600	APAC
Europe, Middle East and Africa	1540.9525	600	EMEA

If you are already locked onto an Atlas signal, you will need to break the lock on the Atlas satellite before JFREQ will manually tune to your new signal. To do this, either disconnect the antenna momentarily, cycling power to the receiver, issuing the JBOOT,OMNI command, or block signal to the antenna physically, for example by covering it with something metallic.

Topic Last Updated: v1.04 / May 29, 2012

JFORCEAPP Command

Command Type

General Operation and Configuration Commands

Description

Force an application to be used in a multi-application (MFA)

Note: This command is not saved; it is only for the current session.

Command Format

Force an application to be used

```
$JFORCEAPP,app<CR><LF>
```

where 'app' is one of the following applications:

- AUTO = allow automatic selection of the application in the MFA (default setting)
- RTK
- SBAS

Query the current setting

```
$JFORCEAPP<CR><LF>
```

Receiver Response

Response to issuing command to force an application to be used

$>

Response to querying the current setting

If currently set to SBAS the response is:

```
$>JFORCEAPP,SBAS
```

Example

Additional Information

Topic Last Updated: v1.05 / January 18, 2013

JGEO Command

Command Type

SBAS

Description

Display information related to the current frequency of SBAS and its location in relation to the receiver's antenna

Command Format

$JGEO[,ALL]<CR><LF>

where ',ALL' is an optional field that displays information for all SBAS satellites (including those not being used)

Receiver Response

$>JGEO,SENT=1575.4200,USED=1575.4200,PRN=prn,LON=lon,EL=ele,AZ=az

where:

Response Component	Description
JGEO	Message header
Sent=1575.4200	Frequency sent to the digital signal processor
Used=1575.4200	Frequency currently used by the digital signal processor
PRN=prn	WAAS satellite PRN number
Lon=-lon	Longitude of the satellite
El=ele	Elevation angle from the receiver antenna to the WAAS satellite, reference to the horizon
AZ=az	Azimuth from the receiver antenna to the WAAS satellite, reference to the horizon

Example

To display information related to the current frequency of SBAS issue the following command:
$JGEO[,ALL]<CR><LF>

The response is then:
$>JGEO,SENT=1575.4200,USED=1575.4200,PRN=122,LON=-54,EL=9.7,AZ=114.0

To display information for dual SBAS satellites issue the following command:
$JGEO[,ALL]<CR><LF>

The response is:
$>JGEO,SENT=1575.4200,USED=1575.4200,PRN=122,LON=-54,EL=9.7,AZ=114.0

$>JGEO,SENT=1575.4200,USED=1575.4200,PRN=134,LON=178,EL=5.0,AZ=252.6

The first line of output is identical to the output from the first JGEO query above; however, the second line of output provides information on the WAAS satellite not being currently used. Both lines of output follow the same format.

Additional Information

Topic Last Updated: v1.02 / January 25, 2011

JI Command

Command Type	General Operation and Configuration
Description	Display receiver information, such as its serial number and firmware version
Command Format	`$JI<CR><LF>`

Receiver Response

`$>JI,SN,FLT,HW,PROD,SDATE,EDATE,SW,DSP<CR><LF>`

where:

Response Component	Description
SN	Serial number of the GPS engine
FLT	Fleet number
HW	Hardware version
PROD	Production date code
SDATE	Subscription begin date when runningAtlas application; not applicable when running all other applications
EDATE	Subscription expiration date when running Atlas application **OR** receiver subscription code when running all other applications (see Interpreting the $JI and $JK 'Date'/Subscription Codes to determine the meaning of the subscription code)
SW	Application software version number
DSP	DSP version (only valid for Atlas applications)

Example

From a Crescent Vector:
`$>JI,452204,1,7,02122009,01/01/1900,01/01/3007,1.5Pa,46`

From a Crescent with Atlas:
`$>JI,883765,1,7,12052010,01/06/1980,06/30/2011,4.9Pa,11`

Additional Information

Topic Last Updated: v1.02 / January 25,2011

JK Command

Command Type	General Operation and Configuration
Description	Subscribe the receiver to various options, such as higher update rates, e-Dif (or base station capability) or L-Dif or Query for the current subscription expiration date when running Atlas application or the receiver subscription code when running all other applications
Command Format	Subscribe the receiver to specific options `$JK,x…<CR><LF>` where 'x…' is the subscription key provided by Hemisphere GNSS and is 10 characters in length Query the current setting `$JK<CR><LF>`
Receiver Response	Response to issuing command to subscribe `$>` Response to querying the current setting when running Atlas applications `$>JK,EndDate,1HzOnly` where: • 'EndDate' is the subscription end date • '1HzOnly' has a value of 1 if the receiver is limited to 1 Hz output (if the receiver is subscribed to a minimum of 10 Hz output this field is omitted) Response to querying the current setting when running all other applications `$>JK,DateCode,SubscriptionCode,DowngradeCode` where: • 'DateCode' indicates your subscription information (compare last four digits of Date Code to determine your subscription and see the Example section below and the examples in Understanding Additive Codes) • 'SubscriptionCode' is the hex equivalent of the Date Code • 'DowngradeCode' is the output rate in Hertz indicating a downgrade from the default of 10 Hz (if 1, 2 or 5 does not appear the output rate is the default 10 Hz)
Example	If you query the receiver for the current setting when running Atlas applications the response will appear similar to the following: `$>JK,06/30/2011,0`

If you query the receiver for the current setting when running any other application the response will appear similar to the following (Crescent Vector example response shown). Example shows no downgrade code (using default output rate of 10 Hz).

```
$>JK,01/01/3007,7
```

Additional Information	Interpreting the $JI and $JK 'Date'/Subscription Codes

Last Updated: v1.06 / March 10, 2015

JK,SHOW Command

Command Type	General Operation and Configuration
Description	contain authorization information
Command Format	`$JK,SHOW<CR><LF>`
Receiver Response	`$>JK,SHOW,0,SUBOPT,ENDDATE,0,OPT=,SUBSCRIPTION DESCRTIPTION,<CR><LF>`

where:

Response Component	Description
0	UNKNOWN
SUBOPT	Subscription code (see Interpreting the $JI and $JK 'Date'/Subscription Codes to determine the meaning of the subscription code)
END DATE	The subscription end date
0	UNKNOWN
OPT=	
Subscription Description	X HZ The maximum data rate . EDIF Supports EDIF function . RTK Supports RTK function. BASE Supports RTK base function. RAW_DATA Supports the RAW data output . L2_L5 Supports other frequencies besides L1. MULTI_GNSS Supports other satellite system besides GPS. BEIDOUB3 Supports B3 frequencie. ATLAS_LBAND Supports receive ATLAS/China CM signal . ATLAS_Xcm The most accurate accuracy by ATLAS/China CM.

Example :

```
$>JK,SHOW,0,157F,12/31/2016,0,OPT=,20HZ,EDIF,RTK,BASE,RAW_DATA,L2_L5,
MULTI_ GNSS,BEIDOUB3,ATLAS_LBAND,ATLAS_30cm
```

Additional Information Interpreting the $JI and $JK 'Date'/Subscription Codes

Topic Last Updated: v1.07 / Octoter 13, 2016

JLBEAM Command

Command Type L-Band

Description Display the information of each spot beam currently in use by the Atlas receiver

Command Format

```
$JLBEAM<CR><LF>
```

Receiver Response

```
$>JLBEAM,Sent freq,Used freq,Baud xxx,Geo xxx (1)
$>JLBEAM,freq1,lon1,lat1,baud1,satlon1
(2)
.
.
.
$>JLBEAM,freqn,lonn,latn,baudn,satlonn
```

where:

Response Component	Description
"Sent" freq	Frequency sent to the digital signal processor (DSP)
"Used" freq	Frequency currently being used by the digital signal processor (DSP)
"Baud" xxxx	Currently used baud rate of the acquired signal
"Geo" xxx	Currently used satellites longitude (in degrees)

The output second line components are described in the following table:

Response Component	Description
freq	Frequency of the spot beam
lon	Longitude of the center of the spot beam (in degrees)
lat	Latitude of the center of the spot beam (in degrees)
baud	Baud rate at which this spot beam is modulated
satlon	Satellites longitude (in degrees)

Example

```
$>JLBEAM,Sent 1551.4890,Used 1551.4890,Baud 1200,Geo -101

$>JLBEAM,1556.8250,-88,45,1200,(-101)

$>JLBEAM,1554.4970,-98,45,1200,(-101)

$>JLBEAM,1551.4890,-108,45,1200,(-101)
```

```
$>JLBEAM,1531.2300,25,50,1200,(16)
$>JLBEAM,1535.1375,-75,0,1200,(-98)
$>JLBEAM,1535.1375,-165,13,1200,(-98)
$>JLBEAM,1535.1525,20,6,1200,(25)
$>JLBEAM,1558.5100,135,-30,1200,(160)
$>JLBEAM,1535.1375,90,15,1200,(109)
$>JLBEAM,1535.1375,179,15,1200,(109)
```

Additional Information

Topic Last Updated: v1.00 / August 11, 2010

JLIMIT Command

Command Type	General Operation and Configuration
Description	Set the threshold of estimated horizontal performance for which the DGPS position LED is illuminated or query the current setting.
Command Format	Set the threshold of estimated horizontal performance `$JLIMIT,limit<CR><LF>` where 'limit' is the new limit in meters Query the current setting `$JLIMIT<CR><LF>`
Receiver Response	Receiver response when setting the threshold of estimated horizontal performance $> Receiver response when querying the current threshold of estimated horizontal performance `$>JLIM,RESID,LIMIT` where 'LIMIT' is the limit in meters
Example	To set the threshold to 5 m issue the following command: `$JLIMIT,5<CR><LF>` If you then query the receiver with `$JLIMIT<CR><LF>` the response is: `$JLIM,RESID,5.00`
Additional Information	The default value for this parameter is a conservative 10.00 m. The status of this command is also output in the JSHOW message.

Topic Last Updated: v1.02 / January 25, 2011

JLXBEAM Command

Command Type [L-Band]

Description Display spot beam debug information

Command Format

```
$JLXBEAM<CR><LF>
```

Receiver Response

```
$>JLBEAMEX
$> Beam:1,DDSfreq1,symbol1,lon1,lat1,lonrad1,latrad1,beamrot1,satlon1,*

$> Beam:2,DDSfreq2,symbol2,lon2,lat2,lonrad2,latrad2,beamrot2,satlon2,*


$> Beam:n,DDSfreqn,symboln,lonn,latn,lonradn,latradn,beamrotn,satlonn,*
```

where:

Response Component	Description
DDSfreq	DDS frequency
symbol	Symbol rate used for that particular spot beam
lon	Longitude of the spot beam centroid
lat	Latitude of the spot beam centroid
lonrad	Longitude radius of the spot beam
latrad	Latitude radius of the spot beam
beamrot	Rotation angle of the spot beam
satlon	Longitude of the Atlas satellite
*	Reserved

Example

```
$>JLBEAMEX

$> Beam:22,1535125000,600,-26,40,2,41,0,9999,*

$> Beam:21,1535157500,600,65,30,31,18,-21,64,*
```

```
$> Beam:13,1535185000,1200,136,-25,23,28,-40,144,*

$> Beam:13,1535185000,1200,172,-40,13,26,-26,144,*

$> Beam:24,1557835000,1200,-100,49,6,28,0,-101,*

$> Beam:24,1557835000,1200,-101,66,12,6,0,-101,*

$> Beam:25,1557845000,1200,-74,52,12,30,-30,-101,*

$> Beam:26,1557855000,1200,-122,45,11,30,25,-101,*

$> Beam:8,1535137500,1200,-85,2,30,20,-5,-98,*

$> Beam:8,1535137500,1200,-60,-25,34,36,-20,-98,*

$> Beam:4,1535137500,1200,109,2,14,19,-27,109,*

$> Beam:4,1535137500,1200,140,38,27,51,-56,109,*

$> Beam:7,1537440000,1200,23,-2,29,49,50,25,*

$> Beam:7,1537440000,1200,14,59,41,23,34,25,*

$> Beam:7,1537440000,1200,11,28,17,24,0,25,*
```

Additional Information

Topic Last Updated: v1.02 / January 25, 2011

JMASK Command

Command Type	GPS
Description	Specify the elevation cutoff mask angle for the GPS engine Any satellites below this mask angle will be ignored even if available. The default angle is 5° because satellites available below this angle will have significant tropospheric refraction errors.
Command Format	`$JMASK,e<CR><LF>` where the elevation mask cutoff angle 'e' may be a value from 0 to 60°
Receiver Response	`$>`
Example	To specify the elevation cutoff mask angle to 10° issue the following command: `$JMASK,10<CR><LF>`
Additional Information	To query the receiver for the current setting, issue the JSHOW command.

Topic Last Updated: v1.02 / January 25, 2011

JMODE

JMODE Overview

The JMODE command is used to control various GPS tracking parameters.

Command	Description
JMODE	Query receiver for status of JMODE settings
JMODE,BASE	Enable/disable base mode functionality or query the current setting
JMODE,BDSOFF	Set the receiver to use BDS data in the solution
JMODE,FIXLOC	Set the receiver to not re-average (or re-average) its position or query the current setting
JMODE,FOREST	Enable/disable high gain functionality (for tracking under canopy) or query the current setting
JMODE,GLOFIX	Enable/disable use of RTCM v3 (RTK) GLONASS correctors
JMODE,GLOOFF	Set the receiver to use GLONASS data in the solution
JMODE,GPSOFF	Set the receiver to use GPS data in the solution
JMODE,GPSONLY	Set the receiver to use GPS data in the solution or query the current setting (if GLONASS is available, setting to YES will cause the receiver to only use GPS data)
JMODE,L1ONLY	Set the receiver to use L1 data even if L2 data is available or query the current setting
JMODE,MIXED	Include satellites that do not have DGPS or SBAS corrections in the solution
JMODE,NULLNMEA	Enable/disable output of NULL fields in NMEA 0183 messages when no there is no fix (when position is lost)
JMODE,SBASNORTK	Disable/enable the use of SBAS ranging signals (carrier phase) in RTK
JMODE,SBASR	Enable/disable SBAS ranging or query the current setting
JMODE,STRICTRTK	Use this command to invoke stricter checks on whether RTK fix is declared. Forces float of RTK at 30 seconds of Age-of-Diff
JMODE,SURETRACK	Enable/disable SureTrack functionality (default is enabled) or query the current setting
JMODE,SURVEY	Assure RTK fix is not declared when residual errors exceed 10 cm. Also forces use of GLONASS and prevents SureTrack operation
JMODE,TIMEKEEP	Enable/disable continuous time updating in NMEA 0183 messages when there is no fix (when position is lost) or query the current setting
JMODE,TUNNEL	Enable/disable faster reacquisition after coming out of a tunnel or query the current setting

Topic Last Updated: v1.07 / Octoter 13, 2016

JMODE Command

Command Type	General Operation and Configuration
Description	Query receiver for status of JMODE settings
Command Format	`$JMODE<CR><LF>`
Receiver Response	`$>JMODES[,BASE][,FIXLOC][,FOREST][,GLOFIX][,GPSONLY][,L1ONLY][,MIXED][,NULLNM`
Example	If FOREST and TUNNEL are set to ON and all others (MIXED, NULLNMEA, SBASR, and TIMEKEEP) are set to OFF and you issue `$JMODES,TUNNEL,FOREST` If all features are set to OFF and you issue the JMODE command the receiver response willbe: `$JMODES`
Additional Information	The status of this command is also output in the JSHOW response. For example, if TUNNEL is set to ON and all other JMODE option `$>JSHOW,MODES,TUNNEL`

Topic Last Updated: v1.04 / May 29, 2012

JMODE,BASE Command

Command Type

General Operation and Configuration, Local Differential and RTK Commands

Description

Enable/disable base mode functionality or query the current setting

- If base mode is NO (disabled) and the receiver is receiving RTK corrections, these corrections are echoed out when RTK corrections (ROX, RTCM3, CMR) are requested
- If base mode is YES (enabled), the receiver computes its own corrections, regardless of whether or not it is receiving RTK corrections from another source

Command Format

Enable/disable base mode

To enable base mode:

```
$JMODE,BASE,YES<CR><LF>
```

To disable base mode:

```
$JMODE,BASE,NO<CR><LF>
```

Query the current setting

```
$JMODE,BASE<CR><LF>
```

Receiver Response

Response to issuing command to enable/disable base mode

```
$>
```

Response to querying the current setting

If base mode is currently enabled the response is:

```
$>JMODE,BASE,YES
```

If base mode is currently disabled the response is:

```
$>JMODE,BASE,NO
```

Example

Additional Information

Topic Last Updated: v1.04 / May 29, 2012

JMODE,BDSOFF Command

Command Type	General Operation and Configuration
Description	Set the receiver to use BDS data in the solution
Command Format	Close/Open BDS operation Close BDS operation: `$JMODE,BDSOFF,YES<CR><LF>` Open BDS operation: `$JMODE,BDSOFF,NO<CR><LF>`
Receiver Response	$> Response to issuing command to turn enable/disable BDS operation Response to querying the current setting If BDS operation is currently enabled the response is: `$>JMODE,BDSOFF,YES` If BDS operation is currently disabled the response is: `$>JMODE,BDSOFF,NO`
Additional Information	

Topic Last Updated: v1.07 / Octoter 13, 2016

JMODE,FIXLOC Command

Command Type	General Operation and Configuration
Description	Set the receiver to not re-average (or re-average) its position or query the current setting. $JMODE,FIXLOC,YES assure that the BASE will not re-average its position. Good for permanent installations.
Command Format	Enable/disable position re-averaging To set receiver to <u>not</u> re-average its position: `$JMODE,FIXLOC,YES<CR><LF>` To set receiver to re-average its position: `$JMODE,FIXLOC,NO<CR><LF>` Query the current setting `$JMODE,FIXLOC<CR><LF>`
Receiver Response	Response to issuing command to enable/disable position re-averaging $> Response to querying the current setting If setting is currently enabled (no position re-averaging) the response is: `$>JMODE,FIXLOC,YES` If setting is currently disabled (position re-averaging enabled) the response is: `$>JMODE,FIXLOC,NO`
Example	
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JMODE,FOREST Command

Command Type	General Operation and Configuration
Description	Enable/disable high gain functionality (for tracking under canopy) or query the current setting. This command is useful if you are trying to maximize the likelihood of calculating a position, but are willing to sacrifice accuracy. See also JMODE,MIXED.
Command Format	Enable/disable high gain functionality To enable high gain functionality: `$JMODE,FOREST,YES<CR><LF>` To disable high gain functionality: `$JMODE,FOREST,NO<CR><LF>` Query the current setting `$JMODE,FOREST<CR><LF>`
Receiver Response	Response to issuing command to turn functionality on/off `$>` Response to querying the current setting If high gain functionality is currently enabled the response is: `$>JMODE,FOREST,YES` If high gain functionality is currently disabled the response is: `$>JMODE,FOREST,NO`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JMODE,GLOFIX

Command Type	General Operation and Configuration
Description	Enable/disable use of RTCM v3 (RTK) GLONASS correctors. GLOFIX does not affect CMR or ROX (CMR does not have GLONASS, and ROX correctors are always used regardless of the GLOFIX setting) and SureTrack is automatically used for any satellite that does not have GLONASS correctors.
Command Format	Enable/disable use of RTCM v3 GLONASS correctors To enable use of RTCM v3 GLONASS correctors: `$JMODE,GLOFIX,YES<CR><LF>` To disable use of RTCM v3 GLONASS correctors: `$JMODE,GLOFIX,NO<CR><LF>` Query the current setting `$JMODE,GLOFIX<CR><LF>`
Receiver Response	Response to issuing command to turn functionality on/off $> Response to querying the current setting If use of RTCM v3 GLONASS correctors is currently enabled the response is: `$>JMODE,GLOFIX,YES` If use of RTCM v3 GLONASS correctors is currently disabled the response is: `$>JMODE,GLOFIX,NO`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JMODE,GLOOFF Command

Command Type	General Operation and Configuration
Description	Set the receiver to use GLONASS data in the solution
Command Format	Close/Open GLONASS operation Close GLONASS operation: `$JMODE,GLOOFF,YES<CR><LF>` Open GLONASS operation: `$JMODE,GLOOFF,NO<CR><LF>`
Receiver Response	$> Response to issuing command to turn enable/disable GLONASS operation Response to querying the current setting If GLONASS operation is currently enabled the response is: `$>JMODE,GLOOFF,NO` If GLONASS operation is currently disabled the response is: `$>JMODE,GLOOFF,YES`
Additional Information	

Topic Last Updated: v1.07 / Octoter 13, 2016

JMODE,GPSOFF Command

Command Type	General Operation and Configuration
Description	Set the receiver to use GPS data in the solution or query the current setting
Command Format	Close/Open GPS operation Close GPS operation: `$JMODE,GPSOFF,YES<CR><LF>` Open GPS operation: `$JMODE,GPSOFF,NO<CR><LF>`
Receiver Response	Response to issuing command to turn enable/disable GPS-only operation $> Response to querying the current setting If GPS-only operation is currently enabled the response is: `$>JMODE,GPSONLY,YES` If GPS-only operation is currently disabled the response is: `$>JMODE,GPSONLY,NO`
Additional Information	

Topic Last Updated: v1.07 / February 16, 2017

JMODE,GPSONLY Command

Command Type General Operation and Configuration

Description Set the receiver to use GPS data in the solution or query the current setting (if GLONASS is available, setting to YES will cause the receiver to only use GPS data)

Command Format

Enable/disable GPS-only operation

Enable GPS-only operation:

```
$JMODE,GPSONLY,YES<CR><LF>
```

Disable GPS-only operation (use GLONASS as well if available):

```
$JMODE,GPSONLY,NO<CR><LF>
```

Query the current setting

```
$JMODE,GPSONLY<CR><LF>
```

Receiver Response

Response to issuing command to turn enable/disable GPS-only operation

```
$>
```

Response to querying the current setting

If GPS-only operation is currently enabled the response is:

```
$>JMODE,GPSONLY,YES
```

If GPS-only operation is currently disabled the response is:

```
$>JMODE,GPSONLY,NO
```

Additional Information

Topic Last Updated: v1.02 / January 25, 2011

JMODE,L1ONLY Command

Command Type	General Operation and Configuration
Description	Set the receiver to use L1 data even if L2 data is available or query the current setting: • When set to YES receiver will use Atlas DGPS service or L1 RTK • When set to NO receiver will use Atlas high precision services or L1/L2 RTK
Command Format	Set receiver to use/not use L1 data even if L2 data is available To use L1 data (even if L2 data is available): `$JMODE,L1ONLY,YES<CR><LF>` To use L2 data if it is available: `$JMODE,L1ONLY,NO<CR><LF>` Query the current setting `$JMODE,L1ONLY<CR><LF>`
Receiver Response	Response to issuing command to turn functionality on/off $> Response to querying the current setting If the receiver is currently using L1 data only even if L2 data is available the response is: `$>JMODE,L1ONLY,YES` If the receiver is currently using L2 data if it is available the response is: `$>JMODE,L1ONLY,NO`
Additional Information	

Topic Last Updated: v1.07 / February 16, 2017

JMODE,MIXED Command

Command Type	General Operation and Configuration
Description	Include satellites that do not have DGPS or SBAS corrections in the solution or query the current setting This command is useful if you are trying to maximize the likelihood of calculating a position, but are willing to sacrifice accuracy. See also JMODE,FOREST.
Command Format	To include/exclude satellites without DGPS or SBAS corrections To include satellites without DGPS or SBAS corrections: `$JMODE,MIXED,YES<CR><LF>` To exclude satellites without DGPS or SBAS corrections: `$JMODE,MIXED,NO<CR><LF>` Query the current setting `$JMODE,MIXED<CR><LF>`
Receiver Response	Response to issuing command to include/exclude satellites without DGPS or SBAS corrections `$>` Response to querying the current setting If satellites without differential corrections are currently included the response is: `$>JMODE,MIXED,YES` If satellites without differential corrections are currently excluded the response is: `$>JMODE,MIXED,NO`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JMODE,NULLNMEA Command

Command Type	General Operation and Configuration
Description	Enable/disable output of NULL fields in NMEA 0183 messages when no there is no fix (when position is lost) or query the current setting This only applies to position portion of the messages; it does not affect the time portion of the message. If this setting is disabled and position is lost then the positioning parameters of the message from the most recent known position are repeated (instead of being NULL if enabled).
Command Format	Enable/disable output of NULL fields in NMEA 0183 messages To enable output: `$JMODE,NULLNMEA,YES<CR><LF>` To disable output: `$JMODE,NULLNMEA,NO<CR><LF>` Query the current setting `$JMODE,NULLNMEA<CR><LF>`
Receiver Response	Response to issuing command to enable/disable output of NULL fields in NMEA 0183 messages $> Response to querying the current setting If setting is currently enabled the response is: `$>JMODE,NULLNMEA,YES` If setting is currently disabled the response is: `$>JMODE,NULLNMEA,NO`
Example	If the most recent GPGGA message is as follows: `$GPGGA,220715.00,3333.4254353,N,11153.3506065,W,2,10,1.0,406.614,M,-26.294,M,6.0,1001*70` ...and then position is lost and JMODE,NULLNMEA is set to **NO** the GPGGA message repeats as follows (most recent known values do not change): `$GPGGA,220715.00,3333.4254353,N,11153.3506065,W,2,10,1.0,406.614,M,-26.294,M,6.0,1001*70` For the same message, if position is lost and JMODE,NULLNMEA is set to **YES** the GPGGA message repeats as follows (position parameters are NULL): `$GPGGA,220716.00,,,,,0,,,,M,,M,,*48`
Additional	

Information

Topic Last Updated: v1.03 / January 11, 2012

JMODE,SBASNORTK Command

Command Type General Operation and Configuration

Description Disable/enable the use of SBAS ranging signals (carrier phase) in RTK

Command Format

Disable/enable use of SBAS ranging signals in RTK

To disable use of SBAS ranging signals in RTK:

`$JMODE,SBASNORTK,YES<CR><LF>`

To enable use of SBAS ranging signals in RTK:

`$JMODE,SBASNORTK,NO<CR><LF>`

Query the current setting

`$JMODE,SBASNORTK<CR><LF>`

Receiver Response

Response to issuing command to disable/enable the use of SBAS ranging signals in RTK

`$>`

Response to querying the current setting

If current setting is to disable SBAS ranging the response is:

`$>JMODE,SBASNORTK,YES`

If current setting is to enable SBAS ranging the response is:

`$>JMODE,SBASNORTK,NO`

Example

Additional Information

Topic Last Updated: v1.04 / May 29, 2012

JMODE,SBASR Command

Command Type	General Operation and Configuration
Description	Enable/disable SBAS ranging or query the current setting
Command Format	Enable/disable SBAS ranging To enable SBAS ranging: `$JMODE,SBASR,YES<CR><LF>` To disable SBAS ranging: `$JMODE,SBASR,NO<CR><LF>` Query the current setting `$JMODE,SBASR<CR><LF>`
Receiver Response	Response to issuing command to enable/disable SBAS ranging $> Response to querying the current setting If setting is currently enabled the response is: `$>JMODE,SBASR,YES` If setting is currently disabled the response is: `$>JMODE,SBASR,NO`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JMODE,STRICTRTK Command

Command Type

General Operation and Configuration

Description

Use this command to invoke stricter checks on whether RTK fix is declared. Forces float of RTK at 30 seconds of Age-of-Diff

Command Format

Enable/disable STRICTRTK functionality

To enable STRICTRTK functionality:

`$JMODE,STRICTRTK,YES<CR><LF>`

To disable STRICTRTK functionality:

`$JMODE,STRICTRTK,NO<CR><LF>`

Query the current setting

`$JMODE,SURETRACK<CR><LF>`

Receiver Response $>

Response to issuing command to enable/disable command

Response to querying the current setting

If setting is currently enabled the response is:

`$>JMODE,STRICTRTK,YES`

If setting is currently disabled the response is:

`$>JMODE,STRICTRTK,NO`

Additional Information

This mode is not saved between power cycles..

Topic Last Updated: v1.04 / May 29, 2012

JMODE,SURETRACK Command

Command Type	General Operation and Configuration
Description	Enable/disable SureTrack functionality (default is enabled) or query the current setting
Command Format	Enable/disable SureTrack functionality To enable SureTrack functionality: `$JMODE,SURETRACK,YES<CR><LF>` To disable SureTrack functionality: `$JMODE,SURETRACK,NO<CR><LF>` Query the current setting `$JMODE,SURETRACK<CR><LF>`
Receiver Response	Response to issuing command to enable/disable SureTrack functionality $> Response to querying the current setting If setting is currently enabled the response is: `$>JMODE,SURETRACK,YES` If setting is currently disabled the response is: `$>JMODE,SURETRACK,NO`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JMODE,SURVEY Command

Command Type	General Operation and Configuration
Description	Assure RTK fix is not declared when residual errors exceed 10 cm. Also forces use of GLONASS and prevents SureTrack operation.
Command Format	Enable/disable continuoustime updating To enable this command `$JMODE,SURVEY,YES<CR><LF>` To disable this command: `$JMODE,SURVEY,NO<CR><LF>` Query the current setting `$JMODE,SURVEY<CR><LF>`
Receiver Response	Response to issuing command to enable/disable command `$>` Response to querying the current setting If setting is currently enabled the response is: `$>JMODE,SURVEY,YES` If setting is currently disabled the response is: `$>JMODE,SURVEY,NO`
Additional Information	This mode is not saved between power cycles (for now)..

Topic Last Updated: v1.07 / Octoter 13, 2016

JMODE,TIMEKEEP Command

Command Type	General Operation and Configuration
Description	Enable/disable continuous time updating in NMEA 0183 messages when there is no fix (when position is lost) or query the current setting When position is lost the time is the only parameter in the message that continues to update; all other parameters remain the same.
Command Format	Enable/disable continuous time updating To enable continuous time updating: `$JMODE,TIMEKEEP,YES<CR><LF>` To disable continuous time updating: `$JMODE,TIMEKEEP,NO<CR><LF>` Query the current setting `$JMODE,TIMEKEEP<CR><LF>`
Receiver Response	Response to issuing command to enable/disable continuous time updating $> Response to querying the current setting If setting is currently enabled the response is: `$>JMODE,TIMEKEEP,YES` If setting is currently disabled the response is: `$>JMODE,TIMEKEEP,NO`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JMODE,TUNNEL Command

Command Type	General Operation and Configuration
Description	Enable/disable faster reacquisition after coming out of a tunnel or query the current setting
Command Format	Enable/disable faster reacquisition after coming out of a tunnel To enable faster reacquisition: `$JMODE,TUNNEL,YES<CR><LF>` To disable faster reacquisition: `$JMODE,TUNNEL,NO<CR><LF>` Query the current setting `$JMODE,TUNNEL<CR><LF>`
Receiver Response	Response to issuing command to turn functionality on/off $> Response to querying the current setting If setting is currently enabled the response is: `$>JMODE,TUNNEL,YES` If setting is currently disabled the response is: `$>JMODE,TUNNEL,NO`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JMSG99 Command

Type	Vector
Description	Change the output in the Bin99 message to be from the specified antenna
Format	`$JMSG99,0` where '0' is used view the primary antenna SNR (default) `$JMSG99,1` where '1' is used view the secondary antenna SNR
Receiver Response	`$>`
Other	

Topic Last Updated: v1.06 / March 10, 2015

JNMEA

JNMEA,GGAALLGNSS Command

Command Type	GLONASS
Description	Configure the GGA string to include full GNSS information (the number of used GNSS satellites will be included in the GPGGA message) or query the current setting The GGA message is only supposed to report position and satellite information based on the GPS constellation. The combined constellation position and satellite data should be reported in the GNSS message, but some users with older equipment cannot utilize this message. This command allows users with older equipment that require a GGA message to be able to utilize and take advantage of the larger constellation of GNSS satellites.
Command Format	Include/exclude full GNSS information in GGA string To include full GNSS information in GGA string: `$JNMEA,GGAALLGNSS,YES<CR><LF>` To exclude full GNSS information from GGA string: `$JNMEA,GGAALLGNSS,NO<CR><LF>` Query the current setting `$JNMEA,GGAALLGNSS<CR><LF>`
Receiver Response	Include/exclude full GNSS information in GGA string $> Query the current setting If set to yes, querying the current setting returns the following: `$>JNMEA,GGAALLGNSS,YES` If set to no, querying the current setting returns the following: `$>JNMEA,GGAALLGNSS,NO`
Additional Information	

Topic Last Updated: v1.07 February 16, 2017

JNMEA,PRECISION Command

Command Type	GPS, Local Differential and RTK, L-Band
Description	Specify or query the number of decimal places to output in the GPGGA, GPLL, and GPGNS messages or query the current setting
Command Format	Specify the number of decimal places `$JNMEA,PRECISION,x<CR><LF>` where 'x' specifies the number of decimal places from 1 to 8 Query the current setting `$JNMEA,PRECISION<CR><LF>`
Receiver Response	Specify the precision `$>` Query the current setting `$>JNMEA,PRECISION,x` where 'x' refers to the number of decimal places to output
Additional Information	When using RTK orAtlas high precision services, Hemisphere GNSS recommends you set JNMEA,PRECISION to at least 7 decimal places. High accuracy positioning techniques require at least 7 decimal places to maintain millimeter (mm) accuracy. This command is the same as JNP.

Topic Last Updated: v1.07 / February 16, 2017

JNP Command

Command Type	GPS, Local Differential and RTK, L-Band
Description	Specify or query the number of decimal places to output in the GPGGA, GPGLL, and GPGNS messages or query the current setting
Command Format	Specify the number of decimal places `$JNP,x<CR><LF>` where 'x' specifies the number of decimal places from 1 to 8 Query the current setting `$JNP<CR><LF>`
Receiver Response	Specify the number of decimal places to output $> Query the current setting `$>JNP,x` where 'x' refers to the number of decimal places to output
Additional Information	When using RTK or Atlas high precision services, Hemisphere GNSS recommends you set JNP to at least 7 decimal places. High accuracy positioning techniques require at least 7 decimal places to maintain millimeter (mm) accuracy. This command is the same as JNMEA,PRECISION.

Topic Last Updated: v1.07 / February 16, 2017

JOFF

JOFF Command

Command Type	GPS

Description	Turn off all data messages being output through the current port or other port (or Port C), including any binary messages such as Bin95 and Bin96
Command Format	`$JOFF[,OTHER]<CR><LF>` When you specify the ',OTHER' data field (without the brackets), this command turns off all messages on the other port. There are no variable data fields for this message. You can issue this command as follows to turn off all messages on Port C: $JOFF,PORTC<CR><LF>
Receiver Response	`$>`
Additional Information	To turn off all data messages being output through all ports, including any binary messages such as Bin95 and Bin96, see the JOFF,ALL command

Topic Last Updated: v1.02 / January 25, 2011

JOFF,ALL Command

Command Type	GPS
Description	Turn off all data messages being output through all ports, including any binary messages such as Bin95 and Bin96
Command Format	`$JOFF,ALL<CR><LF>`
Receiver Response	`$>`
Additional Information	To turn off all data messages being output through a single port, including any binary messages such as Bin95 and Bin96, see the JOFF command

Topic Last Updated: v1.02 / January 25, 2011

JPOS Command

Command Type	General Operation and Configuration
Description	Speed up the initial acquisition when changing continents with the receiver or query the receiver for the current position of the receiver (for example, powering up the receiver for the first time in Europe after it has been tested in Canada) The command enables the receiver to begin the acquisition process for the closest SBAS spot beams. This saves some time with acquisition of the SBAS service. However, use of this message is typically not required because of the quick overall startup time of the receiver module.
Command Format	Specify the latitude and longitude `$JPOS,lat,lon<CR><LF>` where both 'lat' and 'lon': • Must be entered in decimal degrees • Do not need to be more accurate than half a degree Query the current setting `$JPOS<CR><LF>`
Receiver Response	Receiver response when specifying the latitude and longitude $> Receiver response when querying the current setting `$>JPOS,LAT,LON`
Additional Information	The status of this command is also output in the JSHOW message.

Topic Last Updated: v1.02 / January 25, 2011

JPPS Command

JPPS,FREQ Command

Command Type	General Operation and Configuration
Description	Specify the pps frequency of the receiver or query the current setting
Command Format	Set the receiver's specific pps frequency `$JPPS,FREQ,r<CR><LF>` where: **'r' = specific pps frequency** Query the current setting `$PPS,FREQ<CR><LF>`
Receiver Response	Response to issuing command $> Response to querying the current setting `$JPPS,FREQ,1.00<CR><LF>`
Example	Issue the following command to set the pps frequency to 2.000 on the current port: `$JPPS,FREQ,2<CR><LF>` ...the response is then: `$>` If you query the current setting now, the response is: `$JPPS,FREQ,2.00<CR><LF>`
Additional Information	This mode is not saved between power cycles

Topic Last Updated: v1.07 / Octoter 13, 2016

JPPS,WIDTH Command

Command Type	General Operation and Configuration
Description	Specify the pps width of the receiver or query the current setting
Command Format	Set the receiver's specific pps width `$JPPS,WIDTH,r<CR><LF>` where: **'r' = specific pps width** Query the current setting `$PPS,WIDTH<CR><LF>`
Receiver Response	Response to issuing command `$>` Response to querying the current setting `$JPPS,WIDTH,999.996<CR><LF>`
Example	Issue the following command to set the pps width to 2.000 on the current port: `$JPPS,WIDTH,2<CR><LF>` ...the response is then: `$>` If you query the current setting now, the response is: `$JPPS,WIDTH,2.000<CR><LF>`
Additional Information	This mode is not saved between power cycles

Topic Last Updated: v1.07 / Octoter 13, 2016

JPRN,EXCLUDE Command

Note: **For advanced users only. Not required for typical operation.**

Command Type	General Operation and Configuration Commands
Description	**For advanced users only.** Exclude GPS and/or other GNSS satellites from being used in the positioning solution or query the current setting

Command Format

Exclude PRNs from being used in the positioning solution

Exclude GPS and/or other GNSS PRNs:

`$JPRN,EXCLUDE[,GPS,x,x,x…][,GLO,y,y,y…][,GAL,z,z,z…]<CR><LF>`

where:

- 'x,x,x...' represents the GPS PRNs you want to exclude
- 'y,y,y...' represents the GLONASS PRNs you want to exclude
- ‘z,z,z…’ represents the GALILEO PRNs you want to exclude

Exclude no GNSS PRNs:
`$JPRN,EXCLUDE,NONE<CR><LF>`

Exclude no GPS PRNs
`$JPRN,EXCLUDE,GPS,NONE<CR><LF>`

Exclude no GLONASS PRNs:
`$JPRN,EXCLUDE,GLO,NONE<CR><LF>`

Exclude no GALILEO PRNs:
`$JPRN,EXCLUDE,GAL,NONE<CR><LF>`

Query the current setting

Query all excluded PRNs (GPS and GLONASS):

`$JPRN,EXCLUDE<CR><LF>`

Query excluded GPS PRNs:
`$JPRN,EXCLUDE,GPS<CR><LF>`

Query excluded GLONASS PRNs:
`$JPRN,EXCLUDE,GLO<CR><LF>`

Query excluded GALILEO PRNs:
`$JPRN,EXCLUDE,GAL<CR><LF>`

Receiver Response	See Example section below

Example

If you excluded no GPS or GLONASS PRNS and issued the `$JPRN,EXCLUDE,GPS<CR><LF>` command the response is:
`$>JPRN,EXCLUDE,GPS,NONE,GLO,NONE`

If you excluded one GPS PRN (22) and one GLONASS PRN (10) and issued the following commands you would see the following corresponding responses:

- Command: `$JPRN,EXCLUDE,GPS<CR><LF>`
  Response: `$>JPRN,EXCLUDE,GPS,22`
- Command: `$JPRN,EXCLUDE,GLO<CR><LF>`
  Response: `$>JPRN,EXCLUDE,GLO,10`
- Command: `$JPRN,EXCLUDE<CR><LF>`
  Response: `$>JPRN,EXCLUDE,GPS,22,GLO,10`

Additional Information

Topic Last Updated: v1.07 / February 16, 2017

JQUERY

JQUERY,GUIDE Command

Command Type	General Operation and Configuration
Description	Query the receiver for its determination on whether or not it is providing suitable accuracy after both the SBAS and GPS have been acquired (up to five minutes) This feature takes into consideration the download status of the SBAS ionospheric map and also the carrier phase smoothing of the unit.
Command Format	`$JQUERY,GUIDE<CR><LF>`
Receiver Response	If the receiver is ready for use with navigation, or positioning with optimum performance, it returns: `$>JQUERY,GUIDE,YES<CR><LF>` Otherwise, it returns: `$>JQUERY,GUIDE,NO<CR><LF>`
Additional Information	

Topic Last Updated: v1.00 / August 11, 2010

JQUERY,RTKPROG Command

Command Type	Local Differential and RTK
Description	Perform a one-time query of RTK fix progress information
Command Format	`$JQUERY,RTKPROG<CR><LF>` As an alternative you can log this as a message using the JASC,PSAT,RTKPROG command.
Receiver Response	`$>JQUERY,RTKPROG,R,F,N,SS1,SS2,SS3,MASK*CC<CR><LF>` where

Message Component	Description
R	1 = Ready to enter RTK ambiguity fix 0 = Not ready to enter RTK ambiguity fix
F	1 = Receiver running in RTK ambiguity fix mode 0 = Receiver not running in RTK ambiguity fix mode
N	Number of satellites used to fix
SS1	summer-1 SS1 must be significantly larger than SS2 and SS3 to enter R=1 mode
SS2	summer-2
SS3	summer-3
MASK	Bit mask; bits identify which GNSS observables are being received from base recently (1 = GPS, 3 = GPS + GLONASS)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example	`$>JQUERY,RTKPROG,1,1,23,243.3,0.0,0.0,3`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JQUERY,RTKSTAT Command

Command Type [Local Differential and RTK]

Description Perform a one-time query of the most relevant parameters affecting RTK

Command Format

```
$JQUERY,RTKSTAT<CR><LF>
```

As an alternative you can log this as a message using the JASC,PSAT,RTKSTAT command.

Receiver Response

```
$>JQUERY,RTKSTAT,MODE,TYP,AGE,SUBOPT,DIST,SYS,NUM,SNR,RSF,BSF,HAG,
ACCSTAT,SNT
```

where

Message Component	Description
MODE	Mode (FIX,FLT,DIF,AUT,NO)
TYP	Correction type (DFX,ROX,CMR,RTCM3,CMR+,...)
AGE	Age of differential corrections, in seconds
SUBOPT	Subscription code (see Interpreting the $JI and $JK 'Date'/Subscription Codes to determine the meaning of the subscription code)
DIST	Distance to base in kilometers
SYS	Systems in use: • GPS: L1, L2, L5 • GLONASS: G1, G2 • Galileo: E5a, E5b, E5a+b, E6
NUM	Number of satellites used by each system
SNR	Quality of each SNR path, where: • A is > 20 dB • B is > 18 dB • C is > 15 dB • D is <= 15 dB
RSF	Rover slip flag (non zero if parity errors in last 5 minutes, good for detecting jamming and TCXO issues)
BSF	Base slip flag
HAG	Horizontal accuracy guess
ACCSTAT	RTK accuracy status (hex), where:

	• 0x1 = no differential or differential too old, for the application • 0x2 = problems with differential message • 0x4 = horizontal position estimate poor for the application • 0x8 = HDOP high, poor satellite geometry • 0x10 = fewer than 6 L1 sats used • 0x20 = poor L1 SNRs • 0x40 = not in RTK mode • 0x80 = not in RTK mode or RTK only recently solved (< 10 secs ago) • 0x100 = RTK solution compromised, may fail The status message can be any of the above or any combination of the above. For example, a status message of '047' indicates the following: • 0x1 = no differential or differential too old, for the application • 0x2 = problems with differential message • 0x4 = horizontal position estimate poor for the application • 0x40 = not in RTK mode
SNT	Ionospheric scintillation, values are: • 0 (little or no scintillation - does not adversely affect RTK solution) • 1-100 (scintillation detected - adversely affects RTK solution)
<CR>	Carriage return
<LF>	Line feed

Example

```
$>JQUERY,RTKSTAT,FIX,ROX,
1,007F,0.0,(,L1,L2,G1,G2,)(,14,11,9,9,)(,A,A,A,A,),0,1,0.008,000,3
```

Additional Information

Related Commands and Messages

JASC,PSAT,RTKSTAT command
PSAT,RTKSTAT message

Topic Last Updated: v1.05 / January 18, 2013

JQUERY,TEMPERATURE Command

Command Type	General Operation and Configuration
Description	Query the receiver’s temperature
Command Format	: `$JQUERY,TEMPERATURE<CR><LF>`
Receiver Response	`$>JQUERY,TEMPERATURE,51.88`
Additional Information	..

Topic Last Updated: v1.04 / May 29, 2012

JRAD

JRAD Command Overview

This topic provides information related to the NMEA 0183 messages accepted by the receiver's e-Dif application. The following table provides a brief description of the commands supported by the e-Dif application for its control and operation.

Command	Description
JRAD,1	Display the current reference position in e-Dif applications only
JRAD,1,LAT,LON,HEIGHT	Use this command—a derivative of the JRAD,1,P command—when absolute positioning is required in e-Dif applications only
JRAD,1,P	**e-Dif:** Record the current position as the reference with which to compute e-Dif corrections. This would be used in relative mode as no absolute point information is specified. **DGPS Base Station:** Record the current position as the reference with which to compute Base Station corrections in e-Dif applications only. This would be used in relative mode as no absolute point information is specified
JRAD,2	Forces the receiver to use the new reference point (you normally use this command following a JRAD,1 type command)
JRAD,3	Invoke the e-Dif function once the unit has started up with the e-Dif application active, or, update the e-Dif solution (calibration) using the current position as opposed to the reference position used by the JRAD,2 command
JRAD,7	Turn auto recalibration on or off
JRAD,9	Initialize the Base Station feature and use the previously entered point, either with $JRAD,1,P or $JRAD,1,LAT,LON,HEIGHT, as the reference with which to compute Base Station corrections in e-Dif applications only. Use this for both relative mode and absolute mode.
JRAD,10	Specify BDS message to be transmitted by base station

Note: Use the JSAVE command to save changes you need to keep and wait for the $>SAVE COMPLETE response.

Topic Last Updated: v1.07 / Octoter 13, 2016

JRAD,1 Command

Command Type

e-Dif, DGPS Base Station

Description

Display the current reference position in e-Dif applications only

Command Format

```
$JRAD,1<CR><LF>
```

Receiver Response

```
$>JRAD,1,LAT,LON,HEIGHT
```

where:

Command Component	Description
LAT	Latitude of the reference point in decimal degrees
LON	Longitude of the reference point in decimal degrees
HEIGHT	Ellipsoidal height of the reference point in meters

Upon startup of the receiver with the e-Dif application running—as opposed to with the SBAS application—no reference position will be present in memory. If you attempt to query for the reference position, the receiver's response will be:

```
$>JRAD,1,FAILED,PRESENT LOCATION NOT STABLE
```

Example

When you issue the `$JRAD,1` command the response will be similar to the following:

```
$>JRAD,1,51.00233513,-114.08232345,1050.212
```

Additional Information

Topic Last Updated: v1.02 / January 25, 2011

JRAD,1,LAT,LON,HEIGHT Command

Command Type

Dif, DGPS Base Station

e-

Description

Use this command—a derivative of the JRAD,1,P command—when absolute positioning is required in e-Dif applications only

Command Format

```
$JRAD,1,lat,lon,height<CR><LF>
```

where:

Command Component	Description
lat	Latitude of the reference point in decimal degrees
lon	Longitude of the reference point in decimal degrees
height	Ellipsoidal height of the reference point in meters. Ellipsoidal height can be calculated by adding the altitude and the geoidal separation, both available from the GPGGA message. **Example:** `$GPGGA,173309.00,5101.04028,N,11402.38289,W,2,07,1.4, 1071.0,M,- 17.8,M,6.0, 0122*48` ellipsoidal height = 1071.0 + (-17.8) = 1053.2 meters

Both latitude and longitude must be entered as decimal degrees. The receiver will not accept the command if there are no decimal places.

Receiver Response

```
$>JRAD,LAT,LON,HEIGHT
```

Additional Information

Topic Last Updated: v1.00 / August 11, 2010

JRAD,1,P Command

Command Type	Dif, DGPS Base Station

e-

Description	**e-Dif:** Record the current position as the reference with which to compute e-Dif corrections. This would be used in relative mode as no absolute point information is specified. **DGPS Base Station:** Record the current position as the reference with which to compute Base Station corrections in e-Dif applications only. This would be used in relative mode as no absolute point information is specified
Command Format	`$JRAD,1,P<CR><LF>`
Receiver Response	`$>JRAD,1,OK`
Additional Information	

Topic Last Updated: v1.00 / August 11, 2010

JRAD,2 Command

Command Type	e-Dif
Description	Forces the receiver to use the new reference point You normally use this command following a JRAD,1 type command.
Command Format	`$JRAD,2<CR><LF>`
Receiver Response	`$>JRAD,2,OK`
Additional Information	

Topic Last Updated: v1.00 / August 11, 2010

JRAD,3 Command

Command Type	e-Dif
Description	This command has two primary purposes. • To invoke the e-Dif function once the unit has started up with the e-Dif application active • To update the e-Dif solution (calibration) using the current position as opposed to the reference position used by the JRAD,2 command
Command Format	`$JRAD,3<CR><LF>`
Receiver Response	If the receiver has tracked enough satellites for a long enough period before you issue this command, it will respond with the following. (The tracking period can be from 3 to 10 minutes and is used for modeling errors going forward. `$>JRAD,3,OK<CR><LF>` If the e-Dif algorithms do not find sufficient data, the receiver responds with: `$>JRAD,3,FAILED,NOT ENOUGH STABLE SATELLITE TRACKS`
Additional Information	If you receive the failure message after a few minutes of operation, try again shortly after until you receive the “OK” acknowledgement message. The e-Dif application begins operating as soon as the $>JRAD,3,OK message has been received; however, a you will still need to define a reference position for e-Dif unless relative positioning is sufficient for any needs.

Topic Last Updated: v1.00 / August 11, 2010

JRAD,7 Command

Command Type	e-Dif
Description	Turn auto recalibration on or off
Command Format	`$JRAD,7,n` **where 'n' is the auto-recalibration variable (0 = Off or 1 = On, 0 is the default)**
Receiver Response	`$>JRAD,7,OK`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JRAD,9 Command

Command Type	DGPS Base Station
Description	Initialize the Base Station feature and use the previously entered point, either with $JRAD,1,P or $JRAD,1,LAT,LON,HEIGHT, as the reference with which to compute Base Station corrections in e-Dif applications only. Use this for both relative mode and absolute mode.
Command Format	To initialize/turn off base station mode To initialize base station mode and use stored coordinates: `$JRAD,9,1,1<CR><LF>` To turn off base station mode: `$JRAD,9,0<CR><LF>`
Receiver Response	`$>JRAD,9,OK` (same response for turning base station mode on or off)
Additional Information	The $JASC,RTCM,1 command must be sent to the receiver to start outputting standard RTCM corrections.

Topic Last Updated: v1.04 / May 29, 2012

JRAD,10 Command

Command Type	DGPS Base Station
Description	Specify BDS message to be transmitted by base station
Command Format	Specify BDS message to be transmitted by base station `$JRAD,10,1` Specify BDS message to be not transmitted by base station `$JRAD,10,0`
Receiver Response	`$>JRAD,10,OK` (same response for specify BDS to be transmitted or not)
Additional Information	

Topic Last Updated: v1.07 / Octoter 13, 2016

JRAIM Command

Command Type RAIM

Description Specify the parameters of the RAIM scheme that affect the output of the PSAT,GBS message or query the current setting

Command Format

Specify the parameters of the RAIM scheme

```
$JRAIM,hpr,probhpr,probfalse<CR><LF>
```

where:

Command Component	Description
hpr	Horizontal Protection Radius: notification in the PSAT,GBS message that the horizontal error has exceeded this amount will be received. The acceptable range for this value is 1 to 10,000 m. The default is 10 m.
probhpr	Maximum allowed probability that the position computed lies outside the HPR. The acceptable range for this value is 0.001% to 50%. The default is 5%.
probfalse	Maximum allowed probability that there is a false alarm (that the position error is reported outside the of the HPR, but it is really within the HPR). The acceptable range for this value is 0.001% to 50%. The default is 1%.

Query the current setting

```
$JRAIM
```

Receiver Response

Response to issuing command to specify RAIM scheme parameters

$>

Response to querying the current setting

```
$>JRAIM,HPR,probHPR,probFALSE
```

Example To specify the RAIM scheme parameters as HPR = 8 m, probHPR = 2%, and probFALSE = 0.5% issue the following command:

```
$JRAIM,8,2,0.5<CR><LF>
```

If you then query the receiver for the RAIM scheme issue the following command:

```
$JRAIM<CR><LF>
```

...and the response will be:

```
$>JRAIM,8.00,2.0000,0.5000
```

Additional The purpose of the probability of false alarm is to help make a decision on whether to declare a fault or

Information	warning in an uncertain situation. The philosophy is to only issue a fault if the user is certain (to within the probability of a false alarm) that the protection radius has been exceeded, else issue a warning.

Topic Last Updated: v1.02 / January 25, 2011

JRELAY Command

Command Type	General Operation and Configuration
Description	Send user-defined text out of a serial port
Command Format	`$JRELAY,PORTx,msg<CR><LF>` • 'x' = destination port where the message (MSG) will be sent • 'msg' = message to be sent
Receiver Response	`$>`
Example	Example 1: *Command* `$JRELAY,PORTA,HELLO\nTHERE\n<CR><LF>` *Response* `HELLO` `THERE` `$>` **Example 2:** **The following commands apply to the A101 and A325 antennas. You can configure the A101 and A325 through the serial ports using these commands.** • Configure the setup and output of tilt commands as follows (note that all commands are preceded with $JRELAY,PORTC, to direct them through internal Port C): `$JRELAY,PORTC,$JTILT,CALIBRATE[,RESET]` Output the tilt offset values for the X and Y axes. If performing a reset, ensure the A101/A325 is on a flat surface. o `$JRELAY,PORTC,$JTILT,TAU[,value]` Output the filter constant for tilt value smoothing. o `$JRELAY,PORTC,$JTILT,COMPENSATION[,[ON\|OFF],[height offset]]` Turn positioning tilt compensation on/off (currently only the GPGGA data log is supported for tilt compensated position output). o `$JRELAY,PORTC,$JASC,GPGGA,rate[,port]`

Turn tilt compensated GPGGA message on.

- `$JRELAY,PORTC,$JTILT,COGBIAS[,value]`
  Set a COG bias to be used in the tilt compensation algorithms (for use when the A101/A325 is not mounted with the connector facing the forward direction of travel).
- `$JRELAY,PORTC,$JASC,INTLT,rate[,port]`
  or
  `$JRELAY,PORTC,$JASC,PSAT,INTLT,rate[,port]`
  Log tilt information from the A101/A325

- Set/query the receiver mode—serial or NMEA2000 (commands must be sent over Port A):

  `$JRELAY,PORTC,$JQUERYMODE`
  Query the receiver for the current mode
  - `$JRELAY,PORTC,$JSERIALMODE`
    Set the receiver mode to serial
  - `$JRELAY,PORTC,$JN2KMODE`
    Set the receiver mode to NMEA2000

Additional Information

Topic Last Updated: v1.02 / January 25, 2011

JRESET Command

Command Type General Operation and Configuration

Description Reset the receiver to its default operating parameters by:

- Turning off outputs on all ports
- Saving the configuration
- Setting the configuration to its defaults (in following table)

Configuration	Setting
Elev Mask	5
Residual limit	10
Alt aiding	None
Age of Diff	45 minutes
Air mode	Auto
Diff type	Default for app
NMEA precision	5 decimals
COG smoothing	None
speed smoothing	None
WAAS	UERE thresholds

Command Format

```
$JRESET[,x]<CR><LF>
```

where ',x' is an optional field:

- When set to ALL does everything $JRESET does, plus it clears almanacs
- When set to BOOT does everything $JRESET,ALL does, plus clears use of the real-time clock at startup, clears use of backed-up ephemeris and almanacs, and reboots the receiver when done

Receiver Response

```
$JRESET

$> Saving Configuration. Please Wait...

$>

$> Save Complete
```

Additional Information	**CAUTION:** $JRESET clears all parameters. For the V101 Series and the LV101 you will have to issue the $JATT, FLIPBRD,YES command to properly redefine the circuitry orientation inside the product once the receiver has reset. Failure to do so will cause radical heading behavior.

Topic Last Updated: v1.00 / August 11, 2010

JRTCM3

JRTCM3,ANTNAME Command

Command Type Local Differential and RTK

Description Specify the antenna name that is transmitted in various RTCM3 messages from the base

Command Format

Specify the antenna name

$JRTCM3,ANTNAME,name

where name must be an antenna name from the following list:

http://www.ngs.noaa.gov/ANTCAL/LoadFile?file=ngs08.003

Query the current setting

$JRTCM3,ANTNAME<CR><LF>

Receiver Response

Response to issuing command to specify the antenna name

$>

Response to querying the current setting

$JRTCM3,ANTNAME,name

where name is the previously specified antenna name

Example

To specify the antenna name as a Hemisphere GNSS A42 antenna (HEMA42), issue the following command:

$JRTCM3,ANTNAME,HEMA42<CR><LF>

If you then issue $JRTCM3,ANTNAME<CR><LF> to query the current setting the response is:

$>JRTCM3,ANTNAME,HEMA42<CR><LF>

Additional Information

See JRTCM3,NULLANT for information on setting the antenna name to a null value (no name)

Topic Last Updated: v1.06 / March 10, 2015

JRTCM3,EXCLUDE

Command Type [Local Differential and RTK](#)

Description Specify RTCM3 message types to not be transmitted (excluded) by base station

Command Format

Specify the RTCM3 messages to not be transmitted

```
$JRTCM3,EXCLUDE[,1004][,1005][,1006][,1007][,1008][,1012][,1033][,1104]
[,4011][,MSM3][,MSM4]<CR><LF>
```

Query the current setting

```
$JRTCM3,EXCLUDE<CR><LF>
```

Receiver Response

Response to issuing command to exclude specific RTCM3 messages from being transmitted

```
$>
```

Response to querying the current setting

```
$JRTCM3,EXCLUDE[,MSG1][,MSG2]...[,MSGn]<CR><LF>
```

where MSG1 through MSGn represent each included message type to not be transmitted (excluded)

Example Assume all available RTCM3 messages are included (1004, 1005, 1006, 1007, 1008, 1012, 1033). You then issue the following command to exclude message types 1004, 1006, and 1012:

```
$JRTCM3,EXCLUDE,1004,1006,1012<CR><LF>
```

If you then issue `$JRTCM3,EXCLUDE<CR><LF>` to query the current setting the response is:

```
$>JRTCM3,EXCLUDE,1004,1006,1012<CR><LF>
```

Correspondingly, if you issue `$JRTCM3,INCLUDE<CR><LF>` to query the current setting for included messages the response is:

```
$>JRTCM3,INCLUDE,1005,1007,1008,1033<CR><LF>
```

Additional Information See [JRTCM3,INCLUDE](#) for more information on including RTCM3 messages for transmission

Topic Last Updated: v1.07 / Octoter 13, 2016

JRTCM3,INCLUDE Command

Command Type	Local Differential and RTK
Description	Specify RTCM3 message types to be transmitted by base station
Command Format	Specify the RTCM3 messages to be transmitted `$JRTCM3,INCLUDE[,1004][,1005][,1006][,1007][,1008][,1012][,1033][,1104][,4011][,MSM3][,MSM4]<CR><LF>` Query the current setting `$JRTCM3,INCLUDE<CR><LF>`
Receiver Response	Response to issuing command to include specific RTCM3 messages to be transmitted $> Response to querying the current setting `$JRTCM3,INCLUDE[,MSG1][,MSG2]...[,MSGn]<CR><LF>` where MSG1 through MSGn represent each included message type to be transmitted
Example	Assume none of the available RTCM3 messages are included (1004, 1005, 1006, 1007, 1008, 1012, 1033). You then issue the following command to include message types 1004, 1006, and 1012 `$JRTCM3,INCLUDE,1004,1006,1012<CR><LF>` If you then issue `$JRTCM3,INCLUDE<CR><LF>` to query the current setting the response is: `$>JRTCM3,INCLUDE,1004,1006,1012<CR><LF>`
Additional Information	See JRTCM3,EXCLUDE for more information on including RTCM3 messages for transmission

Topic Last Updated: v1.07 / Octoter 13, 2016

JRTCM3,NULLANT Command

Command Type	Local Differential and RTK
Description	Specify the antenna name as null (no name) that is transmitted in various RTCM3 messages from the base
Command Format	Specify the antenna name as null `$JRTCM3,NULLANT<CR><LF>`
Receiver Response	Response to issuing command to exclude specific RTCM3 messages from being transmitted `$>`
Example	Assume you previously specified the antenna name as a Hemisphere GNSS A42 antenna (HEMA42). If you issue `$JRTCM3,ANTNAME<CR><LF>` to query the current setting the response is: `$>JRTCM3,ANTNAME,HEMA42<CR><LF>` Now send the following command to specify the antenna name as null (no name): `$>JRTCM3,NULLANT<CR><LF>` If you then issue `$JRTCM3,ANTNAME<CR><LF>` to query the current setting the response is: `$>JRTCM3,ANTNAME,<CR><LF>`
Additional Information	See JRTCM3,ANTNAME for information on specifying the antenna name as something other than null

Topic Last Updated: v1.06 / March 10, 2015

JRTK

JRTK Command Overview

The JRTK commands are used to define or query RTK settings.

Command	Description
JRTK,1	Show the receiver’s reference position (can issue command to base station or rover)
JRTK,1,LAT,LON,HEIGHT	Set the receiver’s reference position to the coordinates you enter (can issuecommand to base station or rover)
JRTK,1,P	Set the receiver’s reference coordinates to the current calculated position if you donot have known coordinates for your antenna location (can issue command to base station or rover)
JRTK,5	Show the base station’s transmission status for RTK applications (can issuecommand to base station)
JRTK,5,Transmit	Suspend or resume the transmission of RTK (can issue command to base station)
JRTK,6	Display the progress of the base station (can issue command to base station)
JRTK,12	Disable or enable the receiver to go into fixed integer mode (RTK) vs. float mode (L-Dif) - can issue command to rover
JRTK,17	Display the transmitted latitude, longitude, and height of the base station (can issue command to base station or rover)
JRTK,18	Display the distance from the rover to the base station, in meters (can issue command to rover)
JRTK,18,BEARING	Display the bearing from the base station to the rover, in degrees (can issue command to rover)
JRTK,18,NEU	Display the distance from the rover to the base station and the delta North, East, and Up, in meters (can issue command to rover)
JRTK,28	Set the base station ID transmitted in ROX/DFX/CMR/RTCM3 messages (can issue command to base station)

Topic Last Updated: v1.03 / January 11, 2012

JRTK,1 Command

Command Type Local Differential and RTK

Description Show the receiver's reference position (can issue command to base station or rover)

Command Format

```
$JRTK,1<CR><LF>
```

Receiver Response

```
$JRTK,1,LAT,LON,HEIGHT
```

where

Command Component	Description
LAT	Latitude of the reference point in decimal degrees
LON	Longitude of the reference point in decimal degrees
HEIGHT	You must enter HEIGHT as ellipsoidal height in meters. Ellipsoidal height can be calculated by adding the altitude and the geoidal separation, both available from the GPGGA message. **Example:** `$GPGGA,173309.00,5101.04028,N,11402.38289,W,2,07,1.4,1071.0,M,- 17.8,M,6.0, 0122*48` ellipsoidal height = 1071.0 + (-17.8) = 1053.2 meters

Example

```
$>JRTK,1,33.55679117,-111.88955483,374.600
```

Additional Information

Topic Last Updated: v1.02 / January 25, 2011

JRTK,1,LAT,LON,HEIGHT Command

Command Type	Local Differential and RTK
Description	Set the receiver's reference position to the coordinates you enter (can issue command to base station or rover)
Command Format	`$JRTK,1,lat,lon,height<CR><LF>`

where:

Command Component	Description
lat	Latitude of the reference point in decimal degrees
lon	Longitude of the reference point in decimal degrees
height	You must enter HEIGHT as ellipsoidal height in meters. Ellipsoidal height can be calculated by adding the altitude and the geoidal separation, both available from the GPGGA message. **Example:** `$GPGGA,173309.00,5101.04028,N,11402.38289,W,2,07,1.4,1071.0, M,- 17.8,M,6.0, 0122*48` ellipsoidal height = 1071.0 + (-17.8) = 1053.2 meters

Note: You must enter both latitude and longitude in decimal degrees; the receiver will not accept the command if there are no decimal places.

Receiver Response	`$>`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JRTK,1,P Command

Command Type	Local Differential and RTK
Description	Set the receiver's reference coordinates to the current calculated position if you do not have known coordinates for your antenna location (can issue command to base station or rover)
Command Format	`$JRTK,1,P<CR><LF>`
Receiver Response	`$>`
Additional Information	If you have known coordinates for your antenna location, use the JRTK,1,LAT,LON,HEIGHT command to enter the latitude and longitude (in decimal degrees) and the ellipsoidal height (in meters).

Topic Last Updated: v1.02 / January 25, 2011

JRTK,5 Command

Command Type	Local Differential and RTK
Description	Show the base station's transmission status for RTK applications (can issue command to base station)
Command Format	`$JRTK,5<CR><LF>`
Receiver Response	If transmission status is suspended, response is as follows: `$>JRTK,6` If transmission status is not suspended, response is as follows: `$>JRTK,5,1`
Additional Information	Also see the JRTK,6 command.

Topic Last Updated: v1.02 / January 25, 2011

JRTK,5,Transmit Command

Command Type	Local Differential and RTK
Description	Suspend or resume the transmission of RTK (can issue command to base station)
Command Format	`$JRTK,5,transmit<CR><LF>` where "transmit" is 0 (suspend) or 1 (resume)
Receiver Response	If the transmission status is not suspended and you issue the following command to suspend: `$JRTK,5,0<CR><LF>` the response is as follows: `$>JRTK,5,OK` Similarly, if the transmission status is suspended and you issue the following command to resume: `$JRTK,5,1<CR><LF>` the response is again as follows: `$>JRTK,5,OK`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JRTK,6 Command

Command Type [Local Differential and RTK]

Description Display the progress of the base station (can issue command to base station)

Command Format

```
$JRTK,6<CR><LF>
```

Receiver Response

```
$JRTK,6,TimeToGo,ReadyTransmit,Transmitting
```

where

Response Component	Description
TimeToGo	Seconds left until ready to transmit RTK
ReadyTransmit	Non zero when configured to transmit and ready to transmit RTK on at least one port. It is a bit mask of the transmitting port, with bit 0 being port A, bit 1 being port B, and bit 2 being port C. It will be equal to "Transmitting" unless transmission has be suspended with $JRTK,5,0.
Transmitting	Non-zero when actually transmitting RTK on at least one port. It is a bit mask of the transmitting port, with bit 0 being port A, bit 1 being port B, and bit 2 being port C.

Example

If the receiver is not ready to transmit:

```
$>JRTK,6,263,0,0
```

If the receiver is currently transmitting on Port B:

```
$>JRTK,6,0,2,2
```

Additional Information

Topic Last Updated: v1.02 / January 25, 2011

JRTK,12 Command

Warning! **Hemisphere GPS recommends that only advanced users employ this command.**

Command Type	Local Differential and RTK
Description	Disable or enable the receiver to go into fixed integer mode (RTK) vs. float mode (L-Dif) - can issue command to rover **Note:** Requires RTK rover subscription
Command Format	`$JRTK,12,x` where 'x' is: • 1 = Allow RTK (recommended, and the default) • 0 = Do not allow RTK, stay in L-Dif
Receiver Response	`$>`
Additional Information	In high multipath conditions it may be desirable to prevent the rover from obtaining a fixed position. Using $JRTK,12,0 while logging position data is useful for determining the level of multipath present.

Topic Last Updated: v1.02 / January 25, 2011

JRTK,17 Command

Command Type	Local Differential and RTK
Description	Display the transmitted latitude, longitude, and height of the base station (can issue command to base station or rover)
Command Format	`$JRTK,17<CR><LF>`
Receiver Response	`$>JRTK,17,lat,lon,height`
Example	`$>JRTK,17,33.55709242,-111.88916894,380.534`
Additional Information	Format is similar to JRTK,1,LAT,LON,HEIGHT

Topic Last Updated: v1.02 / January 25, 2011

JRTK,18 Command

Command Type	Local Differential and RTK
Description	Display the distance from the rover to the base station, in meters (can issue command to rover)
Command Format	`$JRTK,18<CR><LF>`
Receiver Response	`$>JRTK,18,d` • 'd' is the baseline distance in meters • 'm' indicates the units are meters
Example	`$>JRTK,18,13154.520`
Additional Information	

Topic Last Updated: v1.03 / January 11, 2012

JRTK,18,BEARING Command

Command Type	Local Differential and RTK
Description	Display the bearing from the base station to the rover, in degrees (can issue command to rover)
Command Format	`$JRTK,18,BEARING<CR><LF>`
Receiver Response	`$>JRTK,18,b` • 'b' is the bearing from base to rover in degrees • 'd' indicates the units are degrees
Example	`$>JRTK,18,20.014`
Additional Information	

Topic Last Updated: v1.03 / January 11, 2012

JRTK,18,NEU Command

Command Type	Local Differential and RTK
Description	Display the distance from the rover to the base station and the delta North, East, and Up, in meters (can issue command to rover)
Command Format	`$JRTK,18,NEU<CR><LF>`
Receiver Response	`$>JRTK,18,d,X,Y,Z` where • 'd' is the baseline distance in meters • 'm' indicates the units are meters • 'X' is the North delta, in meters • 'Y' is the East delta, in meters • 'Z' is the Up delta, in meters
Example	`$>JRTK,18,13154.509,12360.045,4502.139,33.739`
Additional Information	

Topic Last Updated: v1.03 / January 11, 2012

JRTK,28 Command

Command Type	Local Differential and RTK
Description	Set the base station ID transmitted in ROX/DFX/CMR/RTCM3 messages (can issue command to base station), where: • Default is 333 • Range is 0-4095 (except for CMR which is 0-31)
Command Format	Set the base station ID `$JRTK,28,baseid<CR><LF>` where 'baseid' is the base station ID Query the current setting `$JRTK,28<CR><LF>`
Receiver Response	`$>`
Example	To set the base station ID to 123 issue the following command: `$JRTK,28,123<CR><LF>` If the base station ID is 333 and you issue the `$JRTK,28<CR><LF>` query the response is: `$>JRTK,28,333`
Additional Information	

Topic Last Updated: v1.02 / January 25, 2011

JSAVE Command

Command Type	General Operation and Configuration
Description	Send this command after making changes to the operating mode of the receiver
Command Format	`$JSAVE<CR><LF>`
Receiver Response	`$> SAVING CONFIGURATION. PLEASE WAIT...` then `$> Save Complete`
Additional Information	Ensure that the receiver indicates that the save process is complete before turning the receiver off or changing the configuration further. No data fields are required. The receiver indicates that the configuration is being saved and indicates when the save is complete.

Topic Last Updated: v1.00 / August 11, 2010

JSHOW

JSHOW Command

Command Type	General Operation and Configuration
Description	Query the current operating configuration of the receiver
Command Format	`$JSHOW<CR><LF>`

Receiver Response

Use the JSHOW command to provide a complete response from the receiver.

Example (number in parentheses corresponds to line number in table following the response):

```
$>JSHOW,BAUD,9600 (1)
$>JSHOW,BAUD,9600,OTHER (2)
$>JSHOW,BAUD,9600,PORTC (3)
$>JSHOW,ASC,GPGGA,1.0,OTHER (4)
$>JSHOW,ASC,GPVTG,1.0,OTHER (5)
$>JSHOW,ASC,GPGSV,1.0,OTHER (6)
$>JSHOW,ASC,GPGST,1.0,OTHER (7)
$>JSHOW,ASC,D1,1,OTHER (8)
$>JSHOW,DIFF,WAAS (9)
$>JSHOW,ALT,NEVER (10)
$>JSHOW,LIMIT,10.0 (11)
$>JSHOW,MASK,5 (12)
$>JSHOW,POS,51.0,-114.0 (13)
$>JSHOW,AIR,AUTO,OFF (14)
$>JSHOW,FREQ,1575.4200,250 (15)
$>JSHOW,AGE,1800 (16)
```

Description of responses:

Line	Description
1	Current port is set to a baud rate of 9600
2	Other port is set to a baud rate of 9600
3	Port C is set to a baud rate of 9600 (Port C is not usually connected externally on the finished product)
4	GPGGA is output at a rate of 1 Hz from the other port
5	GPVTG is output at a rate of 1 Hz from the other port
6	GPGSV is output at a rate of 1 Hz from the other port
7	GPGST is output at a rate of 1 Hz from the other port
8	D1 is output at a rate of 1 Hz from the other port
9	Current differential mode is WAAS
10	Status of the altitude aiding feature (see the JALT command for information how to set turn altitude aiding on or off)

11	Receiver does not support this feature
12	Elevation mask cutoff angle (in degrees)
13	Current send position used for startup, in decimal degrees
14	Current status of the AIR mode (see the JAIR command for information how to set the AIR mode)
15	Current frequency of the augmentation source in use for the receiver (depending on the configuration of the receiver), followed by the bit rate from the SBAS satellite, and optionally followed by 'AUTO' (only when theAtlas receiver is in 'auto-tune' mode)
16	Current maximum acceptable differential age, in seconds (see the JAGE command for information how to set the differential age)

Example See "Receiver Response"section above

Additional Information

Topic Last Updated: v1.07 / February 16, 2017

JSHOW,ASC Command

Command Type	General Operation and Configuration
Description	Query receiver for current ASCII messages being output
Command Format	`$JSHOW,ASC[,x]<CR><LF>` where x is one of the following: • PORTA • PORTB • PORTC • PORTD • OTHER - displays Whatever port you are connected to you do not need to specify that port. For example, if you connected to Port A, the following two commands result in the same response: `$JSHOW,ASC<CR><LF>` `$JSHOW,ASC,PORTA<CR><LF>`
Receiver Response	*See Example section below*

Example

The first row below shows the response to each individual command for Port A (with and without specifying Port A), Port B, and Port C.

The second row shows the response to the generic $JSHOW command with items similar to the first row responses highlighted.

Command Sent to Receiver	Response
$JSHOW,ASC $JSHOW,ASC,PORTA $JSHOW,ASC,PORTB $JSHOW,ASC,PORTC	$>JSHOW,ASC,RTCM,1 $>JSHOW,ASC,RTCM,1 $>JSHOW,ASC,CMR,1,OTHER $>JSHOW,ASC,D1,1,PORTC
$JSHOW	$>JSHOW,BAUD,19200 $>JSHOW,ASC,GPGNS,1.00 $>JSHOW,ASC,GPGRS,1.00 $>JSHOW,BIN,1,1.00

	$>JSHOW,BIN,2,1.00 $>JSHOW,BIN,89,1 $>JSHOW,BIN,99,1 $>JSHOW,ASC,RTCM,1.0 $>JSHOW,BAUD,19200,OTHER $>JSHOW,ASC,CMR,1,OTHER $>JSHOW,BAUD,57600,PORTC $>JSHOW,ASC,GPGGA,1.00,PORTC $>JSHOW,ASC,GPGSV,1.00,PORTC $>JSHOW,ASC,GLGSV,1.00,PORTC $>JSHOW,BIN,69,1,PORTC $>JSHOW,BIN,100,1,PORTC $>JSHOW,ASC,D1,1,PORTC $>JSHOW,DIFF,RTK $>JSHOW,ALT,NEVER $>JSHOW,LIMIT,10.0 $>JSHOW,MASK,5 $>JSHOW,POS,33.6,-112.2 $>JSHOW,AIR,AUTO,NORM $>JSHOW,SMOOTH,LONG900 $>JSHOW,FREQ,1575.4200,250 $>JSHOW,AGE,2700 $>JSHOW,THISPORT,PORTA $>JSHOW,MODES,FOREST,BASE,GPSONLY,GLOFIX,SURETRACK

Additional Information

Topic Last Updated: v1.04 / May 29, 2012

JSHOW,BIN Command

Command Type	General Operation and Configuration
Description	Query receiver for current Bin messages being output
Command Format	`$JSHOW,BIN<CR><LF>`
Receiver Response	`$>JSHOW,BIN,B1,B1R,B2,B2R...,Bn,BnR` where: • B1 is the first Bin message being output B1R is the rate of B1 • B2 is the second Bin message being output B2R is the rate of B2 . . . . • Bn is the last Bin message being output BnR is the rate of Bn
Example	`$>JSHOW,BIN,B01,1.00,B02,1.00,B69,1,B80,1,B89,1,B99,1`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JSHOW,CONF Command

Command Type General Operation and Configuration

Description Query receiver for configuration settings

Command Format `$JSHOW,CONF<CR><LF>`

Receiver Response `$>JSHOW,CONF,AID,AIDVAL,RES,ELEV,MODE,AGE,DIFF`

where:

Message Component	Description	As Displayed in Example Below This Table
AID	Altitude aiding indicator as set by JALT command: • A = ALWAYS • N = NEVER • S = SOMETIMES • T = SATS	A
AIDVAL	Altitude aiding value as by JALT command: • If AID = N, then AIDVAL = 0.0 • If AID = A, then AIDVAL = height • If AID = S, then AIDVAL = PDOP threshold • If AID = T, then AIDVAL = number of sats	404.2
RES	Residual limit for the $JLIMIT command	10.0
ELEV	Elevation mask cutoff angle (in degrees) as set by JMASK command	5
MODETYPE	AIR mode type, A (AUTO) or M (MANUAL), as set by JAIR command	M
MODE	AIR mode, LOW or HIGH or NORM, as set by JAIR command	LOW
AGE	Maximum acceptable differential age (in seconds)	8100 (259200 is using e-Dif)
DIFF	Current differential mode as set by JDIFF command: • T = THIS PORT • P = PORTC • O (letter) = OTHER PORT	A

	• B = BEACON • W = WAAS • R = RTK • L = LBAND • A = X (e-Dif, where A = AUTO) • N = NONE		

Example `$>JSHOW,CONF,A,404.2,10.0,5,M,LOW,259200,A`

Additional Information

Topic Last Updated: v1.04 / May 29, 2012

JSHOW,GP Command

Command Type	General Operation and Configuration
Description	Query the receiver for each GP message currently being output through the current port and the update rate for that message To see output for other ports you must specify that port or OTHER
Command Format	`$JSHOW,GP[,PORTX][,OTHER]<CR><LF>` where: • ',PORTX' = a port other than the current port, such as Port B or Port C • ',OTHER' = Port B if the current port is Port A, or Port A if the current port is Port B
Receiver Response	`$>JSHOW,M1,M1R,M2,M2R...,Mn,MnR` where: • M1 is the first message being output M1R is the rate of M1 • M1 is the first message being output M1R is the rate of M1 . . . • Mn is the last message being output MnR is the rate of Bn
Example	`$>JSHOW,GP,GGA,1.00,GST,1.00`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

JSHOW,THISPORT Command

Command Type	General Operation and Configuration
Description	Query to determine which receiver port you are connected to
Command Format	$JSHOW,THISPORT<CR><LF>
Receiver Response	$>JSHOW,THISPORT,port where 'port' is the port you are connected to
Example	Response if you are connected to Port B: $>JSHOW,THISPORT,PORTB
Additional Information	See JSHOW for information on displaying more configuration information for a receiver

Topic Last Updated: v1.03 / January 11, 2012

JSIGNAL Command

Command Type

General Operation and Configuration

Description

Set the GNSS signals that the receiver will attempt to track. Specific signals shown here are only valid for receivers supporting the signal in question.

Command Format

Specify the signal(s) to be used

```
$JSIGNAL,INCLUDE[,L1CA][,L1P][,L2P][,L2C][,G1][,G2][,E1BC][,B1][,B2][,B3]
[,E5B][,QZSL1CA][,QZSL2C][,ALL]<CR><LF>
```

Specify the signal(s) NOT to be used

```
$JSIGNAL,EXCLUDE[,L1CA][,L1P][,L2P][,L2C][,G1][,G2][,E1BC][,B1][,B2][,B3]
[,E5B][,QZSL1CA][,QZSL2C][,ALL]<CR><LF>
```

Query the current setting

```
$JSIGNAL,INCLUDE<CR><LF>
```

Receiver Response

Response to issuing command to turn functionality on/off

```
$>
```

Response to querying the current setting

```
$>JSIGNAL,INCLUDE[,L1CA][,L1P][,L2P][,L2C][,G1][,G2][,E1BC][,B1][,B2][,B3]
[,E5B][,QZSL1CA][,QZSL2C]<CR><LF>
```

Additional Information

Topic Last Updated: v1.10 / February 16, 2017

JSMOOTH Command

Command Type GPS

Description

Set the carrier smoothing interval (15 to 6000 seconds) or query the current setting

This command provides the flexibility to tune in different environments. The default for this command is 900 seconds (15 minutes) or LONG. A slight improvement in positioning performance (depending on the multipath environment) may occur if you use either the SHORT (300 seconds) or LONG (900 seconds) smoothing interval.

Command Format

Set the carrier smoothing interval

To set the carrier smoothing interval to a specific number of seconds issue the following command:

```
$JSMOOTH,x<CR><LF>
```

where 'x' is one of the following:

- Number of seconds
- DEFAULT (equals 900 seconds)
  Default for e-Dif is 300 seconds
- SHORT (equals 300 seconds)
- LONG (equals 900 seconds)

Query the current setting

```
$JSMOOTH<CR><LF>
```

Receiver Response

Receiver response when setting the carrier smoothing interval

```
$>
```

Receiver response when querying the current carrier smoothinginterval

```
$>JSMOOTH,x
```

where 'x' is the word 'SHORT' or 'LONG' followed by the number of seconds used:

- SHORT precedes the number of seconds for any setting less than 900 seconds
- LONG precedes the number of seconds for any setting greater than or equal to 900 seconds

Example

To set the carrier smoothing interval to 750 seconds issue the following command:

```
$JSMOOTH,750<CR><LF>
```

...and if you then query the receiver using $JSMOOTH the response is:

```
$JSMOOTH,SHORT750
```

To set the carrier smoothing interval to 300 seconds (5 minutes) issue the following command:

```
$JSMOOTH,SHORT<CR><LF>
```

To set the carrier smoothing interval to 900 seconds (15 minutes) issue the following command:

```
$JSMOOTH,LONG<CR><LF>
```

Additional Information	If you are unsure of the best value for this setting, leave it at the default setting of LONG (900 seconds). The status of this command is also output in the JSHOW message.

Topic Last Updated: v1.04 / May 29, 2012

JSYSVER Command

Note: This command is only for querying a receiver for its boot loader version. Before attempting to actually update boot loader software consult Hemisphere GNSS Technical Support.

Command Type	General Operation and Configuration
Description	Returns the boot loader version from the GPS card
Command Format	`$JSYSVER<CR><LF>`
Receiver Response	`$>SYSVER,v` where 'v' is the boot loader version
Example	Response when the boot loader version is 75 `$>SYSVER,75`
Additional Information	

Topic Last Updated: v1.05 / January 18, 2013

JT Command

Command Type General Operation and Configuration

Description Query the receiver for its GPS engine type

Command Format

```
$JT<CR><LF>
```

Receiver Response

```
$>JT,xxxx
```

where xxxx indicates the GPS engine and mode:

JT Command Response (xxxx)	GPS Engine	Mode
DF2b	Eclipse	WAAS, RTK Base
DF2g	Eclipse	L-band
DF2r	Eclipse	RTK Rover
DF3g	Eclipse II	WAAS, RTK Base
DF3i	Eclipse II	e-Dif
DF3r	Eclipse II	RTK Rover
MF3g	miniEclipse	WAAS, RTK Base
MF3i	miniEclipse	e-Dif
MF3r	miniEclipse	RTK Rover
SX2a	Crescent Vector	WAAS RTK
SX2b	Crescent	Base
SX2g	Crescent	WAAS
SX2i	Crescent	e-Dif
SX2r	Crescent	Rover

Example When you issue the `$JT<CR><LF>`command a typical response may be:

```
$>JT,DF2b,MX31rev=28
```

DF2b indicates an Eclipse receiver with WAAS and RTK Base functionality.

Note: MX31rev=28 is the processor type and only appears as part of the Eclipse receiver response. You can disregard the processor type as the text that precedes it (DF2b in this example) provides the requested information (GPS engine and mode).

Additional Information

Topic Last Updated: v1.03 / January 11, 2012

JTAU

JTAU Command Overview

The JTAU command is used to set the time constants for specific parameters for Crescent, Crescent Vector, and Eclipse products.

Command	Description
JTAU,COG	Set the course over ground time (COG) constant and query the current setting
JTAU,SPEED	Set the speed time constant and query the current setting

Topic Last Updated: v1.00 / August 11, 2010

JTAU,COG Command

Note: **The JATT,COGTAU command provides identical functionality but works only with Crescent Vector products.**

Command Type GPS

Description

Set the course over ground (COG) time constant (0.00 to 3600.00 seconds) or query the current setting

This command allows you to adjust the level of responsiveness of the COG measurement provided in the GPVTG message. The default value is 0.00 seconds of smoothing. Increasing the COG time constant increases the level of COG smoothing.

Command Format

Set the COG time constant

`$JTAU,COG,tau<CR><LF>`

where 'tau' is the new COG time constant that falls within the range of 0.00 to 200.1 seconds

The setting of this value depends upon the expected dynamics of the Crescent. If the Crescent will be in a highly dynamic environment, this value should be set lower because the filtering window would be shorter, resulting in a more responsive measurement. However, if the receiver will be in a largely static environment, this value can be increased to reduce measurement noise.

Query the current setting

`$JTAU,COG<CR><LF>`

Receiver Response

Receiver response when setting the COG time constant

$>

Receiver response when querying the current COG time constant

`$>JTAU,COG,tau<CR><LF>`

Example

To set the COG time constant as 2 seconds issue the following command:

`$JTAU,COG,2<CR><LF>`

Additional Information

You can use the following formula to determine the COG time constant: tau (in seconds) = 10 / maximum rate of change of course (in °/s)

If you are unsure about the best value for this setting, it is best to be conservative and leave it at the default setting of 0.00 seconds.

Topic Last Updated: v1.02 / January 25, 2011

JTAU,SPEED Command

Note: The JATT,SPDTAU command provides identical functionality but works only with Crescent Vector products.

Command Type	GPS
Description	Set the speed time constant (0.00 to 3600.00 seconds) or query the current setting This command allows you to adjust the level of responsiveness of the speed measurement provided in the GPVTG message. The default value is 0.00 seconds of smoothing. Increasing the speed time constant increases the level of speed measurement smoothing.
Command Format	Set the speed time constant `$JTAU,SPEED,tau<CR><LF>` where 'tau' is the new speed time constant that falls within the range of 0.0 to 200.2 seconds The setting of this value depends upon the expected dynamics of the receiver. If the receiver will be in a highly dynamic environment, you should set this to a lower value, since the filtering window will be shorter, resulting in a more responsive measurement. However, if the receiver will be in a largely static environment, you can increase this value to reduce measurement noise. Query the current setting `$JTAU,SPEED<CR><LF>`
Receiver Response	Receiver response when setting the speed time constant $> Receiver response when querying the current speed time constants `$>JTAU,SPEED,tau<CR><LF>`
Example	To set the speed time constant as 4.6 seconds issue the following command: `$JTAU,SPEED,4.6<CR><LF>`
Additional Information	You can use the following formula to determine the COG time constant (Hemisphere GNSS recommends testing how the revised value works in practice): tau (in seconds) = 10 / maximum acceleration (in m/s^2) If you are unsure about the best value for this setting, it is best to be conservative and leave it at the default setting of 0.00 seconds.

Topic Last Updated: v1.06 / March 10, 2015

JWAASPRN Command

Command Type	SBAS
Description	Change the SBAS PRNs in memory or query the receiver for current PRNs in memory Valid PRNs include: • EGNOS (Europe SBAS): 120, 124, 126 • GAGAN (India SBAS): 127 • SDCM (Russia SBAS): 125, 141, 140 • MSAS (Japan SBAS): 129, 137 • WAAS (North America SBAS): 133, 135, 138
Command Format	Change the SBAS PRNs in memory `$JWAASPRN,prn1,prn2,prn3<CR><LF>` where 'prn1' and 'prn2' specify PRNs for Crescent receivers and 'prn3' specifies the additional PRN for Eclipse receivers Query the current setting `$JWAASPRN<CR><LF>`
Receiver Response	Response to issuing command to change PRNs $> Response to querying the current setting `$>JWAASPRN,PRN1,PRN2[,PRN3]`
Example	To change the SBAS PRNs in memory for an Eclipse receiver to WAAS PRNs (133, 135, 138) issue the following command: `$>JWAASPRN,133,135,138<CR><LF>`
Additional Information	You can specify an auto-tune mode to tune to the appropriate SBAS PRNs based on the autonomous GPS position. To auto-tune the PRNs issue the following command: `$JWAASPRN,AUTO` If you then query the receiver for the PRNs the receiver response will show ',AUTO' at the end. For example, if you query the receiver and the PRNs are 133,135, and 138 and autotuning is enabled the response is as follows:

```
$>JWAASPRN,133,135,138,AUTO
```

Topic Last Updated: v1.02 / January 25, 2011

PCSI

PCSI,0 Command (Receiver Help Query command)

Command Type	Beacon Receiver
Description	Hemisphere GNSS proprietary NMEA 0183 query Query the SBX to output a list of available proprietary PCSI commands
Command Format	`$PCSI,0<CR><LF>`
Receiver Response	`$PCSI,ACK,0` `$PCSI,P003-0K,012` `$PCSI,0 ->HELP Msg` `$PCSI,1 ->Status line A,<T>,<S>` `$PCSI,2 ->Status line B,<T>` `$PCSI,3 ->Dump Search,<x>` `$PCSI,4 ->Wipe Search` `$PCSI,5 ->Port Rate,<P0>,<P1>` `$PCSI,6 ->Reset` `$PCSI,7 ->RTCM Mode`
Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

PCSI,1 Command (Status Line A, Channel 0 command)

Command Type
Beacon Receiver

Description
Hemisphere GNSS proprietary NMEA 0183 query
Query the SBX for a selection of parameters related to the operational status of its primary channel

Command Format
`$PCSI,1<CR><LF>`

Receiver Response
`$PCSI,ACK,1`

`$PCSI,CS0,PXXX-Y.YYY,SN,fff.f,M,ddd,R,SS,SNR,MTP,WER,ID,H,T,G`

where:

Response Component	Description
CS0	Channel 0
PXXX-Y.YYY	Resident SBX firmware version
SN	SBX receiver serial number
fff.f	Channel 0 current frequency
M	Frequency mode (A = automatic, M = manual, D = database)
ddd	MSK bit rate
R	RTCM rate mode (A = automatic, M = manual, D = database)
SS	Signal strength
SNR	Signal-to-noise ratio
MTP	Message throughput
WER	Word Error Rate - Percentage of bad 30-bit RTCM words in the last 25 words
ID	Beacon ID to which the receiver's primary channel is tuned
H	Health of the tuned beacon [0-7]
T	$PCSI,1 status output period [0-99]
G	AGC gain in dB (0 to 48 db)

Additional Information
Optionally you can modify the Status Line A query to request the output of the response message once every period at a specified output rate. It has the following format, where 'T' is the output period in seconds:
`$PCSI,1,T<CR><LF>`

The response will be:
`$PCSI,ACK,1`

```
$PCSI,CS0,PXXXY.YYY,SN,fff.f,M,ddd,R,SS,SNR,MTP,WER,ID,H,T,G
```

You can stop the output of the message by either of the following:

- Cycling receiver power
- Issuing the $PCSI,1<CR><LF> query without the output period field

The response message has the same format as discussed above. In addition to this modified version of the Status Line A command, an additional 'S' field may be placed after the 'T' field, resulting in the following command:

```
$PCSI,1,T,S<CR><LF>
```

The 'S' field is not a variable and specifies that the output of the Status Line A message should continue after the power has been cycled. To return the receiver to the default mode (in which message output ceases after receiver power is cycled) send the $PCSI,1<CR><LF> query to the receiver.

You may send the $PCSI,1 query through either serial port for reporting of the full status of the primary receiver channel. The query response is returned to the port from which you issued the command. When querying the primary receiver channel using the secondary serial port, no interruptions in RTCM data output will occur on the primary port provided the SBX has acquired a valid beacon.

The response is different depending on whether you are connected directly to the SBX-4 or not.

- If connected directly (by hardware or JCONN), the response will be both an acknowledgement as well as the full PCSI,1 message.
- If connected through a Crescent receiver (such as the R110) you may see the full PCSI,1 message. Consider PCSI,1,1 to generate periodic output.

Topic Last Updated: v1.06 / March 10, 2015

PCSI,1,1 Command (Beacon Status command)

Command Type

Beacon Receiver

Description

Obtain PCSI,CS0 beacon status data from an SBX engine when interfaced to the receiver Port D. When you send this command through either Port A, B, or C it is automatically routed to Port D. The resulting PCSI,CS0 message is returned to the same port from which the command was sent at the desired rate.

Command Format

```
$PCSI,1,1<CR><LF>
```

Receiver Response

```
$PCSI,CS0,Pxxx-y.yyy,SN,fff.f,M,ddd,R,SS,SNR,MTP,WER,ID,H,T,G
```

where:

Response Component	Description
CS0	Channel 0
PXXX-Y.YYY	Resident SBX firmware version
SN	SBX receiver serial number
fff.f	Channel 0 current frequency
M	Frequency mode (A = automatic, M = manual, D = database)
ddd	MSK bit rate
R	RTCM rate mode (A = automatic, M = manual, D = database)
SS	Signal strength
SNR	Signal-to-noise ratio
MTP	Message throughput
WER	Word Error Rate - Percentage of bad 30-bit RTCM words in the last 25 words
ID	Beacon ID to which the receiver's primary channel is tuned
H	Health of the tuned beacon (0-7)
T	$PCSI,1 status output period (0-99)
G	AGC gain in, dB (0 to 48)

Example

```
$PCSI,CS0,P030-0.000,19001,313.0,D,100,D,18,8,80,0,63,0,1,48
```

Additional Information

Topic Last Updated: v1.04 / May 29, 2012

PCSI,2 Command (Status Line B, Channel 1 command)

Command Type [Beacon Receiver]

Description

Hemisphere GNSS proprietary NMEA 0183 query

Query the SBX to output a selection of parameters related to the operational status of its secondary channel

Command Format

```
$PCSI,2<CR><LF>
```

Receiver Response

```
$PCSI,ACK,2

$PCSI,CS1,PXXX-Y.YYY,SN,fff.f,M,ddd,R,SS,SNR,MTP,WER,ID,H,T
```

where:

Response Component	Description
CS1	Channel 1
PXXX-Y.YYY	Resident SBX firmware version
SN	SBX receiver serial number
fff.f	Channel 1 current frequency
M	Frequency mode (A = automatic, M = manual, D = database)
ddd	MSK bit rate
R	RTCM rate mode (A = automatic, M = manual, D = database)
SS	Signal strength
SNR	Signal to noise ratio
MTP	Message throughput
WER	Word error rate - Percentage of bad 30-bit RTCM words in the last 25 words
ID	Beacon ID to which the receiver’s secondary channel is tuned
H	Health of the tuned beacon (0-7)
T	$PCSI,1 status output period (0-99)

Example

```
$PCSI,ACK,2

$PCSI,CS1,P030-0.004,770737,291.0,D,200,D,-7,2,0,100,1024,8,0
```

Additional Information

Optionally you can modify the Status Line B query to request the output of the response message once every period. It has the following format, where T is the output period inseconds:

```
$PCSI,2,T<CR><LF>
```

The response will:

```
$PCSI,ACK,2
```

```
$PCSI,CS0,PXXX-Y.YYY,SN,fff.f,M,ddd,R,SS,SNR,MTP,WER,ID,H,T
```

The response message has the same format as discussed above. The Status Line B message output cannot be set to remain active after the power of the SBX has been cycled.

The $PCSI,2 query may be sent through the either serial port for reporting of the full status of the secondary receiver channel. The response to the query is returned to the port from which the command was issued. When querying the secondary receiver channel using the secondary serial port, no interruptions in RTCM data output will occur on the primary port provided that SBX has acquired avalid beacon.

Topic Last Updated: v1.06 / March 10, 2015

PCSI,3,1 Command (Receiver Search Dump command)

Command Type

Beacon Receiver

Description

Hemisphere GNSS proprietary NMEA 0183 query

Query the SBX to output the search information used for beacon selection in Automatic Beacon Search mode. The output has three frequencies per line.

Command Format

```
$PCSI,3,1<CR><LF>
```

Receiver Response

```
$PCSI,ACK,3,1

$PCSI,tag1,freq1,ID1,chan1,snr1,ss1,tag2,freq2,ID2,chan2,snr2,ss2,

tag3,freq3,ID3,chan3,snr3,ss3
```

where:

Response Component	Description
tag	Channel number with a range of 1 to 84
freq	Channel frequency (kHz * 10)
ID	Beacon ID
chan	Channel information
snr	SNR (dB)
ss	Signal Strength (dBuV/m)

Example

```
$PCSI,ACK,3,1

$PCSI,01,2835,209,0E,00,-0009,02,2840,339,0E,00,-
0012,03,2845,006,0E,00,0009

$PCSI,04,2850,342,0E,00,-0010,05,2855,547,0E,00,-0005,06,2860,109,0E,00,- 0011

$PCSI,07,2865,188,0E,00,-0007,08,2870,272,0E,00,-0004,09,2875,682,0E,00,- 0006

$PCSI,10,2880,645,0E,00,-0007,11,2885,256,0E,00,-0009,12,2890,000,06,00,- 0012

$PCSI,13,2895,132,0E,00,-0009,14,2900,281,0E,00,-0010,15,2905,634,0E,00,- 0008

$PCSI,16,2910,172,0E,00,-0007,17,2915,006,0E,00,-0009,18,2920,546,0E,00,- 0014

$PCSI,19,2925,358,0E,00,-0008,20,2930,479,0E,00,-0009,21,2935,358,0E,00,-
```

```
0011

$PCSI,22,2940,853,0E,00,-0005,23,2945,588,0E,00,-0015,24,2950,210,0E,00,- 0011

$PCSI,25,2955,000,06,00,-0011,26,2960,663,0E,00,-0010,27,2965,596,0E,00,- 0009

$PCSI,28,2970,000,06,00,-0009,29,2975,917,0E,00,-0009,30,2980,000,06,00,- 0016

$PCSI,31,2985,343,0E,00,-0013,32,2990,546,0E,00,-0010,33,2995,546,0E,00,- 0010

$PCSI,34,3000,172,0E,00,-0014,35,3005,006,0E,00,-
0011,36,3010,1006,0E,00,-0009

$PCSI,37,3015,006,0E,00,-0015,38,3020,300,0E,00,-0013,39,3025,277,0E,00,- 0100

$PCSI,40,3030,479,0E,00,-0010,41,3035,006,0E,00,-0012,42,3040,050,0E,00,- 0008

$PCSI,43,3045,000,06,00,-0014,44,3050,172,0E,00,-0013,45,3055,000,06,00,- 0011

$PCSI,46,3060,000,06,00,-0011,47,3065,000,06,00,-0014,48,3070,000,06,00,- 0010

$PCSI,49,3075,000,06,00,-0012,50,3080,006,0E,00,-0015,51,3085,000,06,00,- 0015

$PCSI,52,3090,300,0E,00,-0007,53,3095,000,06,00,-0013,54,3100,000,06,00,- 0013

$PCSI,55,3105,000,06,00,-0012,56,3110,127,0E,00,-0013,57,3115,000,06,00,- 0012

$PCSI,58,3120,596,0E,00,-0012,59,3125,051,0E,00,-0009,60,3130,000,06,00,- 0011

$PCSI,61,3135,213,0E,00,-0008,62,3140,000,06,00,-0011,63,3145,000,06,00,- 0015

$PCSI,64,3150,302,0E,00,-0008,65,3155,000,06,00,-0009,66,3160,000,06,00,- 0003

$PCSI,67,3165,000,06,00,-0013,68,3170,000,06,00,-
0011,69,3175,612,0E,01,0000

$PCSI,70,3180,000,06,00,-0015,71,3185,000,06,00,-0008,72,3190,000,06,00,- 0009

$PCSI,73,3195,000,06,00,0011,74,3200,1002,0E,01,-0002,75,3205,067,0E,00,- 0008

$PCSI,76,3210,001,0E,00,-0008,77,3215,000,06,00,-0009,78,3220,132,0E,00,- 0009

$PCSI,79,3225,000,06,00,-0010,80,3230,339,0E,00,-0013,81,3235,000,06,00,- 0011
```

```
$PCSI,82,3240,000,06,00,-0010,83,3245,202,0E,00,-0007,84,3250,006,0E,00,- 0002
```

Additional Information	

Topic Last Updated: v1.06 / March 10, 2015

PCSI,3,2 Command (Ten Closest Stations command)

Command Type

Beacon Receiver

Description

Display the ten closestbeacon stations

Command Format

```
$PCSI,3,2<CR><LF>
```

Receiver Response

```
$PCSI,ACK,3,2
$PCSI,3,2,StationID,name,freq,status,time,date,distance,health,WER
$PCSI,3,2, ...
$PCSI,3,2, ...
$PCSI,3,2, ...
$PCSI,3,2, ...
...
```

where:

Response Component	Description
StationID	Specific ID number for beacon stations (appears in the last field of the GPGGA message)
name	Name of station
freq	Frequency, in kHz (scaled by 10), on which the station is transmitting. In the first line of the Example below, 2870 indicates 287.0 kHz.
status	0 (operational), 1 (undefined), 2 (no information), 3 (do not use)
time	Not implemented. Currently displayed at 0
date	Not implemented. Currently displayed at 0
distance	Calculated in nautical miles
health	-1 (not updated), 8 (undefined), 0-7 (valid range)
WER	-1 (not updated), 0-100 (valid range)

Example

```
$PCSI,ACK,3,3
$PCSI,3,2, 849,Polson       MT,2870,0,210,0,0,-1,-1
$PCSI,3,2, 848,Spokane      WA,3160,0,250,0,0,-1,-1
$PCSI,3,2, 907,Richmond     BC,3200,0,356,0,0,-1,-1
```

```
$PCSI,3,2, 888,Whidbey Is.   WA,3020,0,363,0,0,-1,-1
$PCSI,3,2, 887,Robinson Pt.  WA,3230,0,383,0,0,-1,-1
$PCSI,3,2, 874,Billings      MT,3130,0,389,0,0,-1,-1
$PCSI,3,2, 871,Appleton      WA,3000,0,420,0,0,-1,-1
$PCSI,3,2, 908,Amphitrite Pt BC,3150,0,448,0,0,-1,-1
$PCSI,3,2, 886,Fort Stevens  OR,2870,0,473,0,0,-1,-1
$PCSI,3,2, 909,Alert Bay     BC,3090,0,480,0,0,-1,-1
```

Additional Information

Topic Last Updated: v1.04 / May 29, 2012

PCSI,3,3 Command (Station Database command)

Command Type Beacon Receiver

Description Display the contents of the beacon station database

Command Format

```
$PCSI,3,3<CR><LF>
```

Receiver Response

```
$PCSI,ACK,3,3

$PCSI,3,3,IDref1,IDref2,StationID,name,freq,lat,long,datum,status

$PCSI,3,3, ...

$PCSI,3,3, ...

$PCSI,3,3, ...

$PCSI,3,3, ...

...
```

where:

Response Component	Description
IDref1	Beacon reference ID (primary)
IDref2	Beacon reference ID (secondary)
StationID	Specific ID number for beacon stations (appears in the last field of the GPGGA message)
name	Name of station
freq	Frequency, in kHz (scaled by 10), on which the station is transmitting. In the first line of the Example below, 2950 indicates 295.0 kHz.
lat	Scaled by 364 (+ve indicates N and -ve indicates S)
long	Longitude is scaled by 182 (+ve indicates N and -ve indicates S)
datum	1 (NAD83), 0(WGS84)
status	0 (operational), 1(undefined), 2 (no information), 3, (do not use)

Example

```
$PCSI,ACK,3,3

$PCSI,3,3,0282,0283,0891,Level Island  AK,2950,20554,-24221,1,0

$PCSI,3,3,0306,0307,0906,Sandspit      BC,3000,19377,-23991,1,0

$PCSI,3,3,0278,0279,0889,Annette Is.   AK,3230,20044,-23951,1,0

$PCSI,3,3,0300,0301,0909,Alert Bay     BC,3090,18412,-23099,1,0
```

```
$PCSI,3,3,0302,0303,0908,Amphitrite Pt BC,3150,17806,-22850,1,0
$PCSI,3,3,0270,0271,0885,C. Mendocino  CA,2920,14718,-22641,1,0
$PCSI,3,3,0272,0273,0886,Fort Stevens  OR,2870,16817,-22559,1,0
$PCSI,3,3,0304,0305,0907,Richmond      BC,3200,17903,-22407,1,0
$PCSI,3,3,0276,0277,0888,Whidbey Is.   WA,3020,17587,-22331,1,0
...
```

Additional Information

Topic Last Updated: v1.04 / May 29, 2012

PCSI,4 Command (Wipe Search command)

Command Type	Beacon Receiver
Description	Clear search history in Auto mode
Command Format	`$PCSI,4<CR><LF>`
Receiver Response	`$PCSI,ACK,4`
Example	
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

PCSI,5 Command (Set Baud Rates command)

Command Type	Beacon Receiver
Description	Set the baud rate of Port0 and Port1 The baud rate for Port0 is saved for next powerup; however, the baud rate for Port1 always defaults to 4800. **Note:** This command applies when you connect directly to a beacon board, as this command has no effect when a beacon board is integrated with a GNSS receiver.
Command Format	`$PCSI,5,portrate0,portrate1<CR><LF>` where: • portrate0 = desired baud rate for Port0 • portrate1 = desired baud rate for Port1
Receiver Response	$>
Example	
Additional Information	

Topic Last Updated: v1.07 / February 16, 2017

PCSI,6 Command (Reboot command)

Command Type	Beacon Receiver
Description	Reboot SBX receiver
Command Format	`$PCSI,6<CR><LF>`
Receiver Response	See example below
Example	When sending this command your response will appear similar to below: `$PCSI,S/N:00019001` `$PCSI,FCFGcrc,B5E5,CCFGcrc,B5E5,Pass` `$PCSI,FGLBcrc,19BC,CGLBcrc,19BC,Pass` `$PCSI,FLSHcrc,0531 Pass` `$PCSI,FSTAcrc,56C3 Base,2FB2,B077`
Additional Information	

Topic Last Updated: v1.04 / May 29, 2012

PCSI,7 Command (Swap Modes command)

Command Type	Beacon Receiver
Description	Swap modes on the receiver (allowing you to output RTCM and PCSI on the desired ports—Port0 and Port1) **Note:** This command applies when you connect directly to a beacon board, as this command has no effect when a beacon board is integrated with a GNSS receiver.
Command Format	`$PCSI,7,mode<CR><LF>` where mode is: • 1 = PCSI on Port1 and RTCM on Port0 • 2 = PCSI on Port0 and RTCM on Port1
Receiver Response	`$PCSI,ACK,7,mode` For example, when sending the following command... `$PCSI,7,1<CR><LF>` ... the response is: `$PCSI,ACK,7,1`
Example	
Additional Information	

Topic Last Updated: v1.07 / February 16, 2017

Binary Messages Code

Binary Messages Code

This section provides the code for the binary messages that Hemisphere GNSS uses.

```
// BinaryMsg.h

#ifndef    BinaryMsg_H____

#define    BinaryMsg_H____

#ifdef   cplusplus

extern "C" {

#endif

/*

 * Copyright (c) 2006  Hemisphere GNSS and CSI Wireless Inc.,

 * All Rights Reserved.

 *

 * Use and copying of this software and preparation of derivative works

 * based upon this software are permitted. Any copy of this software or

 * of any derivative work must include the above copyright notice, this

 * paragraph and the one after it.  Any distribution of this software or

 * derivative works must comply with all applicable laws.

 *

 * This software is made available AS IS, and COPYRIGHT OWNERS DISCLAIMS

 * ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE

 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

 * PURPOSE, AND NOTWITHSTANDING ANY OTHER PROVISION CONTAINED HEREIN, ANY

 * LIABILITY FOR DAMAGES RESULTING FROM THE SOFTWARE OR ITS USE IS

 * EXPRESSLY DISCLAIMED, WHETHER ARISING IN CONTRACT, TORT (INCLUDING

 * NEGLIGENCE) OR STRICT LIABILITY, EVEN IF COPYRIGHT OWNERS ARE ADVISED

 * OF THE POSSIBILITY OF SUCH DAMAGES.

 */

#if defined(WIN32) || (      ARMCC_VERSION >= 300441)

    #pragma pack(push)

    #pragma pack(4)
```

```
#endif

/**************************************************/

/*   SBinaryMsgHeader                                      */

/**************************************************/

typedef struct

{

    char             m_strSOH[4];        /* start of header ($BIN)          */ unsigned

    short m_byBlockID;                   /* ID of message (1,2,99,98,97,96,95,94,93
or 80 ) */

    unsigned short m_wDataLength;        /* 52 16,304,68,28,300,128,96,56, or 40 */

} SBinaryMsgHeader;

typedef struct

{

    unsigned long       ulDwordPreamble;      /* 0x4E494224    = $BIN   */ unsigned

    long                ulDwordInfo;          /*     0x00340001 or 0x00100002 or
0x01300063 */

} SBinaryMsgHeaderDW;                         /* or 0x00440062 or 0x001C0061 or
0x012C0060 */

                                              /* or 0x0080005F or 0x0060005E or
0x0038005D */

                                              /* or 0x00280050 */

#define BIN_MSG_PREAMBLE        0x4E494224    /* $BIN = 0x4E494224 */

#define BIN_MSG_HEAD_TYPE1      0x00340001    /* 52 = 0x34 */

#define BIN_MSG_HEAD_TYPE2      0x00100002    /* 16 = 0x10 */

#define BIN_MSG_HEAD_TYPE99 0x01300063        /* 99 = 0x63, 304 = 0x130 */

#define BIN_MSG_HEAD_TYPE102 0x01580066 /* 102 = 0x66, 344 = 0x158 */

#define BIN_MSG_HEAD_TYPE101 0x01C00065 /* 101 = 0x65, 448 = 0x1C0 */

#define BIN_MSG_HEAD_TYPE100 0x01040064 /* 100 = 0x64, 260 = 0x104*/

#define BIN_MSG_HEAD_TYPE98 0x00440062        /* 98 = 0x62, 68    = 0x44   */

#define BIN_MSG_HEAD_TYPE97 0x001C0061        /* 97 = 0x61, 28    = 0x1C   */

#define BIN_MSG_HEAD_TYPE96 0x012C0060        /* 96 = 0x60, 300 = 0x12C */

#define BIN_MSG_HEAD_TYPE95 0x0080005F        /* 95 = 0x5F, 128 = 0x80     */

#define BIN_MSG_HEAD_TYPE94 0x0060005E        /* 94 = 0x5E, 96    = 0x60   */

#define BIN_MSG_HEAD_TYPE93 0x0038005D        /* 93 = 0x5D, 56    = 0x38   */
```

```
#define BIN_MSG_HEAD_TYPE91 0x0198005B        /* 91 = 0x5B, 408 = 0x198 = total size in bytes -8 -2 -2*/

#define BIN_MSG_HEAD_TYPE89 0x00500059        /* 89 = 0x59, 80    = 0x50   */

#define BIN_MSG_HEAD_TYPE80 0x00280050        /* 80 = 0x50, 40    = 0x28   */

#define BIN_MSG_HEAD_TYPE76 0x01C0004C        /* 76 = 0x4C, 448 = 0x1C0 = total size in bytes -8 -2 -2*/

#define BIN_MSG_HEAD_TYPE71 0x01C00047        /* 71 = 0x47, 448 = 0x1C0 = total size in bytes -8 -2 -2*/

#define BIN_MSG_HEAD_TYPE16 0x01380010  /* 16 = 0x10, 312 = 0x138 */  //GNSS phase observables

#define BIN_MSG_HEAD_TYPE45 0x0080002D  /* 45 = 0x2D, 128 = 0x80  */  //Galileo subframe words --- similar to GPS

#define BIN_MSG_HEAD_TYPE44 0x0038002C  /* 44 = 0x2C,  56 = 0x38  */  //Galileo time offsets

#define BIN_MSG_HEAD_TYPE61 0x0140003D  /* 61 = 0x3D, 320 = 0x140 */

#define BIN_MSG_HEAD_TYPE62 0x0028003E  /* 62 = 0x3E,  40 =  0x28 */

#define BIN_MSG_HEAD_TYPE65 0x00440041  /* 65 = 0x41,  68 =  0x44 */

#define BIN_MSG_HEAD_TYPE66 0x01600042  /* 66 = 0x42, 352 = 0x160 */

#define BIN_MSG_HEAD_TYPE69 0x012C0045  /* 69 = 0x45, 300 = 0x12C */

#define BIN_MSG_HEAD_TYPE59 0x0100003B  /* 59 = 0x3B, 256 = 0x100 */ //GPS L2C


#define BIN_MSG_HEAD_TYPE10 0x0194000A        /* 10 = 0xA, 404 = 0x194 = total size in bytes -8 -2 -2*/

#if defined(_RXAIF_PLOT_MESSAGES_)

   #define BIN_MSG_HEAD_TYPE11        0x0064000B    /* 11 = 0x0B, 100 = 0x64 = total size(112) in bytes -8 -2 -2*/

#endif

#define BIN_MSG_CRLF             0x0A0D        /* CR LF = 0x0D, 0x0A */

#define CHANNELS_12      12

#define cBPM_SCAT_MEMSIZE 100

#if defined(_RXAIF_PLOT_MESSAGES_)

   #define cBPM_AIFSCAT_MEMSIZE 16

#endif

typedef

union

{

          SBinaryMsgHeader

                         sBytes

        ; SBinaryMsgHeaderDW

              sDWord;
```

```
} SUnionMsgHeader;

/***************************************************/

/*  SBinaryMsg1                                    */

/***************************************************/

typedef struct

{

    SUnionMsgHeader    m_sHead;

    unsigned char    m_byAgeOfDiff;          /* age of differential, seconds (255
max)*/

    unsigned char    m_byNumOfSats;          /* number of satellites used (12 max)
 */

    unsigned short m_wGPSWeek;               /* GPS week */

    double          m_dGPSTimeOfWeek;       /* GPS tow    */

    double          m_dLatitude;            /* Latitude degrees, -90..90 */ double

                    m_dLongitude;           /* Latitude    degrees, -180..180 */

    float           m_fHeight;              /* (m), Altitude ellipsoid */ float

                    m_fVNorth;              /* Velocity north          m/s */

    float           m_fVEast;               /* Velocity eastm/s */

    float           m_fVUp;                 /* Velocity up    m/s */

    float           m_fStdDevResid;         /* (m), Standard Deviation of
  Residuals */

    unsigned short m_wNavMode;

    unsigned short m_wAgeOfDiff;             /* age of diff using 16 bits      */

    unsigned short m_wCheckSum;              /* sum of all bytes of the datalength
*/

    unsigned short m_wCRLF;                  /* Carriage Return Line Feed */

} SBinaryMsg1;                               /* length = 8 + 52 + 2 + 2 = 64 */

/***************************************************/

/*  SBinaryMsg2                                    */

/***************************************************/

typedef struct

{

    SUnionMsgHeader    m_sHead;

    unsigned long    m_ulMaskSatsTracked;     /* SATS Tracked, bit mapped 0..31 */

    unsigned long    m_ulMaskSatsUsed;        /* SATS Used, bit mapped 0..31 */

    unsigned short m_wGpsUtcDiff;             /* GPS/UTC time difference (GPS minus
UTC) */

    unsigned short m_wHDOPTimes10;            /* HDOP          (0.1 units) */ unsigned

    short m_wVDOPTimes10;                     /* VDOP          (0.1 units) */ unsigned
```

```
short m_wWAASMask;                          /* Bits 0-1: tracked sats, Bits 2-3:
```

```
                                                  used sats, Bits 5-9 WAAS PRN 1
minus

                                                  120, Bits 10-14 WAAS PRN 1 minus
120 */

    unsigned short m_wCheckSum;              /* sum of all bytes of the datalength
*/

    unsigned short m_wCRLF;                  /* Carriage Return Line Feed */

} SBinaryMsg2;                               /* length = 8 + 16 + 2 + 2 = 28 */

/****************************************************/

/*   SChannelData                                        */

/****************************************************/

typedef struct

{

    unsigned char m_byChannel;          /* channel number    */

    unsigned char m_bySV;               /* satellite being tracked, 0 == not
tracked   */

    unsigned char m_byStatus;           /* Status bits (code carrier bit frame...)
 */

    unsigned char m_byLastSubFrame; /* last subframe processed*/

    unsigned char m_byEphmVFlag;        /* ephemeris valid flag */

    unsigned char m_byEphmHealth;       /* ephemeris health */ unsigned

    char m_byAlmVFlag;                  /* almanac valid flag */

    unsigned char m_byAlmHealth;        /* almanac health */

    char            m_chElev;           /* elevation angle */ unsigned

    char m_byAzimuth;                   /* 1/2 the Azimuth angle */

    unsigned char m_byURA;              /* User Range Error */

    unsigned char m_byDum;              /* Place Holder */

    unsigned short m_wCliForSNR;        /* code lock indicator for SNR divided by 32
*/

    short           m_nDiffCorr;        /* Differential correction * 100 */

    short           m_nPosResid;        /* position residual * 10 */

    short     m_nVelResid;       /* velocity residual * 10 */   short

    m_nDoppHz;      /* expected doppler in HZ */  short        m_nNCOHz;

    /* track from NCO in HZ */

} SChannelData;    /* 24 bytes */
```

```
/***************************************************/
/*   SChannelL2Data                                */
/***************************************************/
//#if defined(_DUAL_FREQ_) typedef
struct
{
    unsigned char m_byChannel;          /* channel number    */
    unsigned char m_bySV;               /* satellite being tracked, 0 == not
tracked   */
    unsigned char m_byL2CX;             /* Status bits for L2P (code carrier bit
frame...)    */
    unsigned char m_byL1CX;             /* Status bits for L1P (code carrier bit
frame...)    */
    unsigned short m_wCliForSNRL2P; /* code lock indicator for SNR divided by 32 */
    unsigned short m_wCliForSNRL1P; /* code lock indicator for L1P SNR divided by 32 */
    short         m_nC1_L1;          /* C1-L1 in meters * 100 */
    short         m_nP2_C1;          /* P2-C1 in meters * 100 */
    short         m_nP2_L1;          /* P2-L1 in meters * 100 */
    short         m_nL2_L1;          /* L2-L1 in meters * 100 */
    short         m_nP2_P1;          /* P2-P1 in meters * 100 */
    short         m_nNCOHz;          /* track from NCO in HZ */
} SChannelL2Data;     /* 20 bytes */
//#endif
/***************************************************/
/*   SChannelL2CData       for USING_GPSL2CL       */
/***************************************************/
typedef struct
{
    unsigned char m_byChannel;          // channel number
    unsigned char m_bySV;               // satellite being tracked, 0 == not
tracked
    unsigned char m_byL2CX;             // Status bits for L2P (code carrier bit
frame...)
    unsigned char spare1;
```

```
    unsigned short m_wCliForSNRL2C; // code lock indicator for SNR divided by
32

    unsigned short spare2;

    short           m_nL2C_L1Ca;        //L2CL - CA code error     meters * 100

    short           m_nL2C_L2P;         //L2CL - L2P code error meters * 100 short

                    m_nL2_L1;           //L2CL - L1CA phase error meters * 100

    short           m_nL2_L2P;          //L2CL - L2P phase error meters * 100 short

                    spare3;

    short           m_nNCOHz;           // track from NCO in HZ

} SChannelL2CData;     // 20 bytes

/*****************************************************/

/*  SBinaryMsg99                                      */

/*****************************************************/

typedef struct

{

    SUnionMsgHeader    m_sHead;

    unsigned char    m_byNavMode;          /* Nav Mode FIX_NO, FIX_2D, FIX_3D
(high bit =has_diff) */

    char             m_cUTCTimeDiff;       /* whole Seconds between UTC and GPS
  */

    unsigned short m_wGPSWeek;             /* GPS week */

    double           m_dGPSTimeOfWeek;     /* GPS tow    */

    SChannelData  m_asChannelData[CHANNELS_12];  /*  channel  data  */  short

    m_nClockErrAtL1;   /* clock error at L1, Hz */   unsigned short m_wSpare;

                                           /* spare */

    unsigned short m_wCheckSum;            /* sum of all bytes of the datalength
*/

      unsigned short m_wCRLF;                /* Carriage Return Line Feed */

} SBinaryMsg99;                            /* length = 8 + 304 + 2 + 2 = 316 */

#define CHANNELS_SBAS_E        3

/*****************************************************/

/*  SBinaryMsg89    * Supports 3 SBAS Satellites          */

/*****************************************************/

typedef struct
```

```
{
    SUnionMsgHeader    m_sHead;
    long               m_lGPSSecOfWeek;       /* GPS tow integer sec */
    unsigned char      m_byMaskSBASTracked; /* SBAS Sats Tracked, bit mapped0..3
*/
    unsigned char      m_byMaskSBASUSED;      /* SBAS Sats Used, bit mapped 0..3 */
    unsigned short m_wSpare;                  /* spare */
    SChannelData       m_asChannelData[CHANNELS_SBAS_E];      /* SBAS channel data */
    unsigned short m_wCheckSum;               /* sum of all bytes of the datalength
*/
    unsigned short m_wCRLF;                   /* Carriage Return Line Feed */
} SBinaryMsg89;                               /* length = 8 + 80 + 2 + 2 = 92 */
/***************************************************/
/*   SBinaryMsg100                                          */
/***************************************************/
//#if defined(_DUAL_FREQ_) typedef
struct
{
    SUnionMsgHeader    m_sHead;
    unsigned char      m_byNavMode;           /* Nav Mode FIX_NO, FIX_2D, FIX_3D
(high bit =has_diff) */
    char               m_cUTCTimeDiff;        /* whole Seconds between UTC and GPS
  */
    unsigned short m_wGPSWeek;                /* GPS week */
    unsigned long      m_ulMaskSatsUsedL2P; /* L2P SATS Used, bit mapped 0..31 */ double
                       m_dGPSTimeOfWeek;      /* GPS tow    */
    unsigned long      m_ulMaskSatsUsedL1P; /* L1P SATS Used, bit mapped 0..31 */
    SChannelL2Data m_asChannelData[CHANNELS_12]; /* channel data */
    unsigned short m_wCheckSum;               /* sum of all bytes of the datalength
*/
      unsigned short m_wCRLF;                   /* Carriage Return Line Feed */
} SBinaryMsg100;                              /* length = 8 + 260 + 2 + 2 = 272 */
//#endif
/***************************************************/
/*   SBinaryMsg59     for USING_GPSL2CL                   */
```

```
/**************************************************/

typedef struct

{

    SUnionMsgHeader    m_sHead;

    unsigned char    m_byNavMode;           /* Nav Mode FIX_NO, FIX_2D, FIX_3D
(high bit =has_diff) */ //1 byte

    char             m_cUTCTimeDiff;        /* whole Seconds between UTC and GPS
   */                     //1 byte

    unsigned short m_wGPSWeek;              /* GPS week */
                                                 //2 bytes

    unsigned long    m_ulMaskSatsUsedL2P; /* L2P SATS Used, bit mapped 0..31*/
                       //4 bytes

    double           m_dGPSTimeOfWeek;      /* GPS tow   */
                                                //8 bytes

    SChannelL2CData m_asChannelData[CHANNELS_12]; /* channel data*/
                                  //20*12 bytes

    unsigned short m_wCheckSum;             /* sum of all bytes of the datalengtha
*/

    unsigned short m_wCRLF;                 /* Carriage Return Line Feed */

} SBinaryMsg59;                             /* length = 8 + 260 + 2 + 2 = 272 */

/**************************************************/

/*   SSVAlmanData                                      */

/**************************************************/

typedef struct

{

    short           m_nDoppHz;          /* doppler in HZ for stationary receiver */

    unsigned char m_byCountUpdate;      /* count of almanac updates */

    unsigned char m_bySVindex;          /* 0 through 31 (groups of 8)*/

    unsigned char m_byAlmVFlag;         /* almanac valid flag */ unsigned

    char m_byAlmHealth;                 /* almanac health */

    char            m_chElev;           /* elevation angle */ unsigned

    char m_byAzimuth;                   /* 1/2 the Azimuth angle */

} SSVAlmanData;    /* 8 bytes */

/**************************************************/

/*   SBinaryMsg98                                      */

/**************************************************/
```

```
typedef struct
{
    SUnionMsgHeader    m_sHead;

    SSVAlmanData     m_asAlmanData[8];       /* SV data, 8 at a time */
    unsigned char m_byLastAlman;             /* last almanac processed */
    unsigned char m_byIonoUTCVFlag;          /* iono UTC flag */  unsigned
    short m_wSpare;  /* spare */
    unsigned short m_wCheckSum;              /* sum of all bytes of the datalength
*/
    unsigned short m_wCRLF;                  /* Carriage Return Line Feed */
} SBinaryMsg98;                              /* length = 8 + (64+1+1+2) + 2 + 2 = 80
*/
/***************************************************/
/*   SBinaryMsg97                                      */
/***************************************************/
typedef struct
{
    SUnionMsgHeader    m_sHead;
    unsigned long    m_ulCPUFactor;          /* CPU utilization Factor (%=multby
450e-6) */
    unsigned short m_wMissedSubFrame;        /* missed subframes */
    unsigned short m_wMaxSubFramePend;       /* max subframe pending */
    unsigned short m_wMissedAccum;           /* missed accumulations */
    unsigned short m_wMissedMeas;            /* missed measurements */
    unsigned long    m_ulSpare1;             /* spare 1 (zero)*/ unsigned
    long             m_ulSpare2;             /* spare 2 (zero)*/ unsigned
    long             m_ulSpare3;             /* spare 3 (zero)*/ unsigned
    short m_wSpare4; /* spare 4 (zero)*/
    unsigned short m_wSpare5;                /* spare 5 (zero)*/
    unsigned short m_wCheckSum;              /* sum of all bytes of the datalength
*/
      unsigned short m_wCRLF;                  /* Carriage Return Line Feed */
} SBinaryMsg97;                              /* length = 8 + (28) + 2 + 2 = 40 */
/***************************************************/
```

```
/*   SObservations                                          */
/**************************************************/
typedef struct
{
    unsigned long      m_ulCS_TT_SNR_PRN; /* Bits 0-7 PRN (PRN is 0 if no data)
*/
                                          /* Bits 8-15 SNR_value
                                             SNR = 10.0*log10( 0.8192*SNR_value)
*/
                                          /* Bits 16-23 Phase Track Time in units of
                                             1/10 second (range = 0 to 25.5 seconds
                                                        (see next word) */
                                          /* Bits 24-31 Cycle Slip Counter Increments
                                             by 1 every cycle slip with natural roll
                                             over after 255 */
    unsigned long      m_ulDoppler_FL;    /* Bit   0: 1 if Valid Phase, 0
otherwise
                                             Bit   1: 1 if Track Time > 25.5 sec,
                                                       0 otherwise
                                             Bits 2-3: unused
                                             Bits 4-32: Signed (two's compliment)
                                             doppler in units of m/sec x 4096. (i.e.,
                                                       LSB = 1/4096). Range =
                                             +/- 32768 m/sec. Computed as phase
                                             change over 1/10 sec. */
    double             m_dPseudoRange;    /* pseudo ranges (m) */
    double             m_dPhase;          /* phase (m) L1 wave len =
0.190293672798365*/
} SObservations;    /* 24 bytes */
/**************************************************/
/*   SBinaryMsg96                                           */
/**************************************************/
typedef struct
{
```

```
    SUnionMsgHeader  m_sHead;

    unsigned short   m_wSpare1;          /* spare 1 (zero)*/

    unsigned short   m_wWeek;            /* GPS Week Number */

    double           m_dTow;             /* Predicted GPS Time in seconds */

    SObservations    m_asObvs[CHANNELS_12];/* 12 sets of observations*/

    unsigned short   m_wCheckSum;        /* sum of all bytes of the
datalength */

    unsigned short   m_wCRLF;            /* Carriage Return Line Feed*/

} SBinaryMsg96;                          /* length = 8 + (300) + 2 + 2 = 312
*/

/*****************************************************/

/*   SBinaryMsg95                                      */

/*****************************************************/

/* sent only upon command or when values change*/ typedef

struct

{

    SUnionMsgHeader  m_sHead;

    unsigned short   m_wSV;              /* The satellite to which this data
belongs. */

    unsigned short   m_wSpare1;          /* spare 1 (chan number (as zero
9/1/2004)*/

    unsigned long    m_TOW6SecOfWeek;    /* time at which this arrived (LSB =
6sec) */

    unsigned long m_SF1words[10]; /* Unparsed SF 1 message words. */ unsigned long

    m_SF2words[10]; /* Unparsed SF 2 message words. */ unsigned long   m_SF3words[10];

    /* Unparsed SF 3 message words.*/

                                         /* Each of the subframe words
contains

                                            one 30-bit GPS word in the lower

                                            30 bits, The upper two bits are
ignored

                                            Bits are placed in the words from
left to

                                            right as they are received */

    unsigned short   m_wCheckSum;        /* sum of all bytes of the datalength
*/

    unsigned short   m_wCRLF;            /* Carriage Return Line Feed*/
```

```
} SBinaryMsg95;                          /* length = 8 + (128) + 2 + 2 = 140
*/

/************************************************/

/*   SBinaryMsg94                                              */

/************************************************/

/* sent only upon command or when values change */ typedef

struct

{

    SUnionMsgHeader    m_sHead;

    /* Iono parameters. */

    double          m_a0,m_a1,m_a2,m_a3;     /* AFCRL alpha parameters. */

    double          m_b0,m_b1,m_b2,m_b3;     /* AFCRL beta parameters.     */

    /* UTC conversion parameters. */

    double          m_A0,m_A1;               /* Coeffs for determining UTC time. */

    unsigned long   m_tot;                   /* Reference time for A0 & A1, sec of
GPS week. */

    unsigned short m_wnt;                    /* Current UTC reference week number.
*/

    unsigned short m_wnlsf;                  /* Week number when dtlsf becomes
effective. */

    unsigned short m_dn;                     /* Day of week (1-7) when dtlsf
becomes effective. */

    short           m_dtls;                  /* Cumulative past leap seconds. */

    short           m_dtlsf;                 /* Scheduled future leap seconds. */

    unsigned short m_wSpare1;                /* spare 4 (zero)*/

    unsigned short m_wCheckSum;              /* sum of all bytes of the datalength
*/

    unsigned short m_wCRLF;                  /* Carriage Return Line Feed */

} SBinaryMsg94;                          /* length = 8 + (96) + 2 + 2 =      108 */

/************************************************/

/*   SBinaryMsg93                                            */

/************************************************/

/* sent only upon command or when values change */

/* WAAS ephemeris */ typedef

struct
```

```
{
    SUnionMsgHeader     m_sHead;
    unsigned short      m_wSV;              /* The satellite to which this data
belongs. */
    unsigned short      m_wWeek;            /* Week corresponding to m_lTOW*/
    unsigned long       m_lSecOfWeekArrived; /* time at which this arrived (LSB=
1sec) */
    unsigned short      m_wIODE;
    unsigned short      m_wURA;             /* See 2.5.3 of Global Pos Sys Std Pos
Service Spec */
    long m_lTOW;                            /* Sec of WEEK Bit 0 = 1 sec */
    long m_lXG;                             /* Bit 0 = 0.08 m */
    long m_lYG;                             /* Bit 0 = 0.08 m */
    long m_lZG;                             /* Bit 0 = 0.4 m */
    long m_lXGDot;                          /* Bit 0 = 0.000625 m/sec */
    long m_lYGDot;                          /* Bit 0 = 0.000625 m/sec */
    long m_lZGDot;                          /* Bit 0 = 0.004 m/sec */
    long m_lXGDotDot;                       /* Bit 0 = 0.0000125 m/sec/sec */
    long m_lYGDotDot;                       /* Bit 0 = 0.0000125 m/sec/sec */
    long m_lZGDotDot;                       /* Bit 0 = 0.0000625 m/sec/sec */
    short   m_nGf0;                         /* Bit 0 = 2**-31 sec */
    short   m_nGf0Dot;                      /* Bit 0 = 2**-40 sec/sec */
    unsigned short      m_wCheckSum;        /* sum of all bytes of the datalength
*/
    unsigned short      m_wCRLF;            /* Carriage Return Line Feed */
} SBinaryMsg93;                             /* length = 8 + (56) + 2 + 2 = 68 */
/****************************************************/
/*   SBinaryMsg80                                   */
/****************************************************/
typedef struct
{
    SUnionMsgHeader     m_sHead;
    unsigned short m_wPRN;               /* Broadcast PRN */
    unsigned short m_wSpare;             /* spare (zero) */
```

```
    unsigned long    m_ulMsgSecOfWeek;       /* Seconds of Week For Message */

    unsigned long    m_aulWaasMsg[8];        /* Actual 250 bit waas message*/

    unsigned short m_wCheckSum;              /* sum of all bytes of the datalength
*/

    unsigned short m_wCRLF;                  /* Carriage Return Line Feed */

} SBinaryMsg80;                              /* length = 8 + (40) + 2 + 2 = 52 */

/****************************************************/

/*   SMsg91Data                                          */

/****************************************************/

typedef struct

{

    unsigned char    bySV;              /* satellite being tracked, 0 == not
tracked   */

    unsigned char    byStatus;          /* Status bits (code carrier bit frame...)
 */

    unsigned char    byStatusSlave;     /* Status bits (code carrier bit frame...)
 */

    unsigned char    byChannel;         /* Not used */


    unsigned short wEpochSlew;                      /* 20*_20MS_EPOCH_SLEW +
_1MS_EPOCH_SLEW */

    unsigned short wEpochCount;                     /* epoch_count */

    unsigned long    codeph_SNR;                    /* 0-20 = code phase (21 bits), 28-
32 = SNR/4096, upper 4 bits */

    unsigned long    ulCarrierCycles_SNR;           /* 0-23 = carrier cycles, 24-32 =
SNR/4096 lower 8 bits */

    unsigned short wDCOPhaseB10_HalfWarns;          /* 0-11 = DCO phase, 12-14 = Half
Cycle Warn

                                                       15 = half Cycle added */

    unsigned short m_wPotentialSlipCount;           /* potential slip count */

    /* SLAVE DATA */

    unsigned long    codeph_SNR_Slave;                 /* 0-20 = code phase (21
bits), 28-32 = SNR/4096, upper 4 bits */

    unsigned long    ulCarrierCycles_SNR_Slave;        /* 0-23 = carrier cycles, 24-
32 = SNR/4096 lower 8 bits */

    unsigned short wDCOPhaseB10_HalfWarns_Slave; /* 0-11 = DCO phase, 12-14 = Half
Cycle Warn

                                                          15 = half Cycle added */
```

```
unsigned short m_wPotentialSlipCount_Slave;            /* potential slip count */

} SMsg91Data; /* 32 bytes */

/*****************************************************/

  /*  SBinaryMsg91                                        */

  /*  Comment: Transmits data from Takemeas.c             */

  /*            debugging structure.                      */

  /*            Added by bbadke 7/07/2003                 */

  /*****************************************************/

typedef struct

{

    SUnionMsgHeader    m_sHead;                     /* 8 */

    double             m_sec;                       /* 8 bytes */

    int                m_iWeek;                     /* 4 bytes */

    unsigned long      m_Tic;                       /* 4 bytes */

    long               lTicOfWeek;                  /* 4 bytes */

    long               lProgTic;                    /* 4 bytes */ SMsg91Data

                       s91Data[CHANNELS_12];        /* 12*32= 384 bytes */

    unsigned short     m_wCheckSum;                 /* sum of all bytes of the
datalength */

    unsigned short     m_wCRLF;                     /* Carriage Return Line Feed */

} SBinaryMsg91;                                     /* length = 8 + (408) + 2 + 2 =
420 */

/*****************************************************/

/*   SObsPacket                                           */

/*****************************************************/

typedef struct

{

    unsigned long      m_ulCS_TT_W3_SNR;      /* Bits 0-11 (12 bits) =SNR_value

                                                 For L1 SNR = 10.0*log10(
0.1024*SNR_value)

                                                 FOR L2 SNR = 10.0*log10(
0.1164*SNR_value) */

                                              /* Bits 12-14 (3 bits) = 3 bits of
warning
```

```
                                                  for potential 1/2 cycle slips.      A
warning

                                                  exists if any of these bits are
set. */

                                               /* bit 15: (1 bit) 1 if Track Time >
25.5 sec,

                                                                  0 otherwise */

                                               /* Bits 16-23 (8 bits): Track Time in
units

                                                  of 1/10 second (range = 0 to 25.5
seconds) */

                                               /* Bits 24-31 (8 bits) = Cycle Slip
Counter

                                                  Increments by 1 every cycle slip with

                                                  natural roll-over after 255 */

    unsigned long      m_ulP7_Doppler_FL;      /* Bit 0: (1 bit) 1 if Valid Phase, 0
otherwise

                                                  Bit 1-23: (23 bits) =Magnitude of
doppler

                                                      LSB = 1/512 cycle/sec

                                                      Range = 0 to 16384 cycle/sec

                                                  Bit 24: sign of doppler,
1=negative, 0=pos

                                                  Bits 25-31 (7 bits) = upper 7 bits
of the

                                                      23 bit carrier phase.

                                                      LSB = 64 cycles,    MSB = 4096
cycles */

    unsigned long      m_ulCodeAndPhase;       /* Bit 0-15 (16 bits) lower 16 bits of
code

                                                      pseudorange

                                                      LSB = 1/256 meters MSB

                                                      = 128 meters

                                                      Note, the upper 19 bits are
given in

                                                      m_aulCACodeMSBsPRN[] for CA
code

                                                  Bit 16-31 lower 16 bits of the
carrier phase,
```

```
                                          7 more bits are in
m_ulP7_Doppler_FL
                                   LSB = 1/1024 cycles MSB
                                   = 32 cycles */

} SObsPacket;    /* 12 bytes , note: all zero if data not available */

/* A NOTE ON DECODING MESSAGE 76
 * Notation: "code" -- is taken to mean the PseudoRange derived from code
phase.
 *           "phase" -- is taken to mean range derived from carrier phase.
 *                      This will contain cycle ambiguities.
 *
 * Only the lower 16 bits of L1P code, L2P code and the lower 23 bits of
 * carrier phase are provided. The upper 19 bits of the L1CA code are found
 * in m_aulCACodeMSBsPRN[].  The upper 19 bits of L1P or L2P must be derived
 * using the fact that L1P and L2P are within 128 meters of L1CA.   To
 * determine L1P or L2P, use the lower 16 bits provided in the message and
 * set the upper bits to that of L1CA.  Then add or subtract one LSB of the
 * upper bits (256 meters) so that L1P or L2P are within 1/2 LSB (128 meters)
 * of the L1CA code.
 *     The carrier phase is in units of cycles, rather than meters,
 * and is held to within 1023 cycles of the respective code range.   Only
 * the lower 16+7=23 bits of carrier phase are transmitted in Msg 76.
 * In order to determine the remaining bits, first convert the respective
 * code range (determined above) into cycles by dividing by the carrier
 * wavelength.  Call this the "nominal reference phase". Next extract the 16
 * and 7 bit blocks of carrier phase from Msg 76 and arrange to form the lower
 * 23 bits of carrier phase.  Set the upper bits (bit 23 and above) equal to
 * those of the nominal reference phase.  Then, similar to what was done for
 * L1P and L2P, add or subtract the least significant upper bit (8192 cycles)
 * so that carrier phase most closely agrees with the nominal reference phase
 * (to within 4096 cycles).
 */
```

```
#define CHANNELS_12_PLUS (CHANNELS_12+2)                    /* up to two SBAS
satellites */

#define CHANNELS_L1_E       (CHANNELS_12+CHANNELS_SBAS_E) /* All L1 (including SBAS
satellites) */

/**************************************************/

/*   SBinaryMsg76                                       */

/**************************************************/

typedef struct

{

    SUnionMsgHeader  m_sHead;

    double           m_dTow;                    /* GPS Time in seconds */

    unsigned short   m_wWeek;                   /* GPS Week Number */

    unsigned short   m_wSpare1;                 /* spare 1 (zero)*/

    unsigned long    m_ulSpare2;                /* spare 2 (zero)*/

    SObsPacket       m_asL2PObs[CHANNELS_12];   /* 12 sets of L2(P)
observations */

    SObsPacket       m_asL1CAObs[CHANNELS_L1_E];    /* 15 sets of L1(CA)
observations */

    unsigned long    m_aulCACodeMSBsPRN[CHANNELS_L1_E]; /* array of 15words.

                                                   bit 7:0 (8 bits) =
satellite PRN, 0

                                                   if no satellite

                                                   bit 12:8 (5 bits) = spare

                                                   bit 31:13 (19 bits) = upper
19 bits

                                                   of L1CA   LSB = 256 meters

                                                             MSB = 67108864
meters */

    unsigned long    m_auL1Pword[CHANNELS_12];    /* array of 12 words relating to
L1(P) code.

                                                  Bit 0-15 (16 bits) lower 16
bits of the

                                                  L1P code pseudo range.

                                                  LSB = 1/256 meters MSB =

                                                  128 meters

                                                  Bits 16-27 (12 bits) = L1P
SNR_value
```

```
0.1164*SNR_value)                                           SNR = 10.0*log10(

                                                            If Bits 16-27 all zero, no
L1P track

                                                            Bits 28-31 (4 bits) spare */

    unsigned short     m_wCheckSum;                    /* sum of all bytes of the
datalength */

    unsigned short     m_wCRLF;                        /* Carriage Return Line Feed */

} SBinaryMsg76;                                        /* length = 8 + (448) + 2 + 2 =
460 */

/****************************************************/

/*   SMsg71DataL1                                           */

/****************************************************/

typedef struct

{

    unsigned char    bySV;                         /* satellite being tracked, 0 == not
tracked        */

    unsigned char    byStatus;                     /* Status bits (code carrier bit
frame...)    */

    unsigned char    byStatusL1P;                  /* 0-8 lower 8 bits of L1P
SNR/32768, if zero and

                                                      if upper two bits of
m_wSNR_codeph_L1P are zero

                                                      then L1P is not tracking */

    unsigned char    byStatusL2P;                  /* Status bits (code carrier phase
...)   */

    unsigned short wEpochSlew;                     /* 20*_20MS_EPOCH_SLEW +
_1MS_EPOCH_SLEW */

    unsigned short wEpochCount;                    /* epoch_count */

    unsigned long    codeph_SNR;                   /* 0-20 = code phase (21 bits), 28-
32 = SNR/4096, upper 4 bits */

    unsigned long    ulCarrierCycles_SNR;          /* 0-23 = carrier cycles, 24-32 =
SNR/4096 lower 8 bits */

    unsigned short wDCOPhaseB10_HalfWarns;         /* 0-11 = DCO phase, 12-14 = Half
Cycle Warn

                                                      15 = half Cycle added */

    unsigned short m_wPotentialSlipCount;          /* potential slip count */

} SMsg71DataL1; /* 20 bytes */

/****************************************************/
```

```
/*   SMsg71DataL1PL2P                                        */

/****************************************************/

typedef struct

{

    /* L1P and L2P Data */

 //    unsigned long    codeph_SNR_L1P; NOT USED YET      /* 0-22 = L1 code phase (23
bits), 28-32 = SNR/8192, upper 4 bits */

    unsigned long    codeph_SNR_L2P;                      /* 0-22 = L2P code phase (23
bits), 28-32 = SNR/8192, upper 4 bits */

    unsigned long    ulCarrierCycles_SNR_L2P;             /* 0-23 = carrier cycles,
24-32 = SNR/8192 lower 8 bits */

    unsigned short wDCOPhaseB10_L2P;                      /* 0-11 = DCO phase, 12-15 =
Spare */

    unsigned short m_wSNR_codeph_L1P;                     /* 0-13 = lower 14 bits of
L1P code,    14-15 SNR/32768 Upper 2 bits */

                                                          /* To get full L1P code, use
upper bits form L2P and adjust by

                                                             +/- 2**14 if necessary */

} SMsg71DataL1PL2P; /* 12 bytes */

/****************************************************/

  /*  SBinaryMsg71                                           */

  /*  Comment: Transmits data from Takemeas.c                */

  /*           debugging structure for Dual Freq.            */

  /****************************************************/

typedef struct

{

    SUnionMsgHeader    m_sHead;                         /* 8 */

    double             m_sec;                           /* 8 bytes */

    int                m_iWeek;                         /* 4 bytes */

    unsigned long      m_Tic;                           /* 4 bytes */

    long               lTicOfWeek;                      /* 4 bytes */

    long               lProgTic;                        /* 4 bytes */

    SMsg71DataL1PL2P s91L2PData[CHANNELS_12];           /* 12*12 = 144 bytes */

    SMsg71DataL1       s91Data[CHANNELS_12_PLUS];       /* 14*20 = 280 bytes */

    unsigned short     m_wCheckSum;                     /* sum of all bytes of the
datalength */
```

```
    unsigned short     m_wCRLF;                         /* Carriage Return Line Feed
*/

} SBinaryMsg71;                                         /* length = 8 + (448) + 2 + 2
= 460 */

////////////////////////////////////////////////////

//   SBinaryMsg10

//   Comment: Transmits scatter plot data from

//             buffacc.c

//

////////////////////////////////////////////////////

enum eBIN10_TYPE {eBIN10_GPSL1CA=0,eBIN10_GPSL1P,eBIN10_GPSL2P,

                     eBIN10_GLONASSL1,eBIN10_GLONASSL2,eBIN10_GPSL2CL,eBIN10_GPSL5
Q};

typedef struct

{

    SUnionMsgHeader  m_sHead;                       // 8 bytes

    unsigned short m_awScatterPlotDataI[cBPM_SCAT_MEMSIZE]; //100*2 = 200 bytes unsigned

    short  m_awScatterPlotDataQ[cBPM_SCAT_MEMSIZE];  //100*2  =  200  bytes  unsigned  short

    m_wChannel;

    unsigned short m_wSigType;                      // one of eBIN10_TYPE

    unsigned short     m_wCheckSum;                 // sum of all bytes of the
datalength

    unsigned short   m_wCRLF;                       // Carriage Return Line Feed

} SBinaryMsg10;                                     // length = 8 +200 +200 +2 +2 +2
+2 = 416

#if defined(_RXAIF_PLOT_MESSAGES_)

////////////////////////////////////////////////////

//   SBinaryMsg11

//   Comment: Transmits scatter plot data for RXGNSS_AIF statistics

//

//////////////////////////////////////////////////// enum

eBIN11_TYPE {eBIN11_COUNTS=0,eBIN11_VALUES};

typedef struct

{

    SUnionMsgHeader  m_sHead;                       // 8 bytes
```

```
    unsigned short     m_awScatterPlotDataValues[cBPM_AIFSCAT_MEMSIZE];          //16*2
= 32 bytes

    unsigned short     m_awScatterPlotDataCntMag[cBPM_AIFSCAT_MEMSIZE];          //16*2
= 32 bytes

    unsigned short     m_awScatterPlotDataCntDCoff[cBPM_AIFSCAT_MEMSIZE]; //16*2
= 32 bytes

    unsigned short     m_wChannel;               // aif_sel 0: AIF_A, 1: AIF_B, ...

    unsigned short     m_wSigType;               // one of eBIN11_TYPE

    unsigned short     m_wCheckSum;              // sum of all bytes of the
datalength

    unsigned short     m_wCRLF;                  // Carriage Return Line Feed

} SBinaryMsg11;                                  // length = 8 +32 +32 +32 +2 +2 +2
+2 = 112

#endif

/****************************************************/

/*   SGLONASSChanData                                          */

/****************************************************/ typedef

struct

{

       unsigned char m_bySV;                  /* Bit (0-6) = SV slot, 0 == not tracked

                                             * Bit 7 = Knum flag

                                             * = KNum+8 if bit 7 set

                                             */

       unsigned char m_byAlm_Ephm_Flags;       /* ephemeris and almanac status flags */

                                            /* bit 0: Ephemeris available but timed
out

                                             * bit 1: Ephemeris valid

                                             * bit 2: Ephemeris health OK

                                             * bit 3: unused

                                             * bit 4: Almanac available

                                             * bit 5: Almanac health OK

                                             * bit 6: unused

                                             * bit 7: Satellite doesn't exist

                                             */

    unsigned char m_byStatus_L1;            /* Status bits (code carrier bit
frame...)    */
```

```
    unsigned char m_byStatus_L2;           /* Status bits (code carrier bit
frame...)    */

    char            m_chElev;              /* elevation angle */ unsigned

    char m_byAzimuth;                      /* 1/2 the Azimuth angle */

    unsigned char m_byLastMessage;         /* last message processed */

    unsigned char m_bySlip01;              /* cycle slip on chan 1 */

    unsigned short m_wCliForSNR_L1;        /* code lock indicator for SNR divided by
32 */

    unsigned short m_wCliForSNR_L2;        /* code lock indicator for SNR divided by
32 */

    short           m_nDiffCorr_L1;        /* Differential correction * 100 */

    short           m_nDoppHz;             /* expected doppler in HZ at glonass L1
*/

    short           m_nNCOHz_L1;           /* track from NCO in HZ */

    short           m_nNCOHz_L2;           /* track from NCO in HZ */

    short           m_nPosResid_1;         /* position residual 1 * 1000 */

    short           m_nPosResid_2;         /* position residual 2 * 1000 */

} SGLONASSChanData;     /* 24 bytes */

/**************************************************/

/*   SBinaryMsg69                                        */

/**************************************************/

typedef struct

{

    SUnionMsgHeader       m_sHead;

    long                  m_lSecOfWeek;        /* tow   */

    unsigned short        m_wL1usedNavMask;    /* mask of L1 channels used in nav
solution */

    unsigned short        m_wL2usedNavMask;    /* mask of L2 channels used in nav
solution */

    SGLONASSChanData      m_asChannelData[CHANNELS_12]; /* channel data 12X24 = 288 */

    unsigned short        m_wWeek;             /* week */

    unsigned char         m_bySpare01;         /* spare 1 */

    unsigned char         m_bySpare02;         /* spare 2 */

    unsigned short        m_wCheckSum;         /* sum of all bytes of the datalength
*/
```

```
    unsigned short       m_wCRLF;             /* Carriage Return Line Feed */

} SBinaryMsg69;                               /* length = 8 + 300 + 2 + 2 = 312 */

/****************************************************/

/*   SMsg61Data                                                  */

/****************************************************/

typedef struct

{

    unsigned char    bySV;                 /* satellite slot    0 == not tracked    */

    unsigned char    byStatusL1;           /* Status bits (code carrier bit
frame...)    */

    unsigned char    byStatusL2;           /* Status bits (code carrier bit
frame...)    */

    unsigned char    byL1_L2_DCO;          /* 0-3 = upper 4 bits of L1 carrier DCO
Phase

                                            * 4-7 = upper 4 bits of L2 carrier DCO
Phase

                                            */

    unsigned short wEpochSlewL1;           /* 0-9 = slew, 0 to 1000 count for ms of
sec

                                            * 10-15 = 6 bits of L1 slip count */

    unsigned short wEpochCountL1;          /* 0-9 = epoch_count, 0 to 1000 count for
ms of sec

                                            * 10-15 = 6 bits of L2 slip count */

    unsigned long    codeph_SNR_L1;        /* 0-20 =    L1 code phase (21 bits =
9+12),

                                            * 21-32 = L1 SNR/4096 (upper 11 of 12
bits) */

    unsigned long    ulCarrierCycles_L1; /* 0-23 = L1 carrier cycles,

                                            * 24-32 = L1 Carrier DCO lower 8 bits
*/

    unsigned long    codeph_SNR_L2;        /* 0-20 =    L2 code phase (21 bits =
9+12),

                                            * 21-32 = L2 SNR/4096 (upper 11 of 12
bits) */

    unsigned long    ulCarrierCycles_L2; /* 0-23 = L2 carrier cycles,

                                            * 24-32 = L2 Carrier DCO lower 8 bits
*/

} SMsg61Data; /* 24 bytes */
```

```
/***************************************************/
  /*   SBinaryMsg61                                 */
  /*   Comment: Transmits data from TakemeasGLONASS.c */
  /*            debugging structure for Dual Freq.  */
  /***************************************************/
typedef struct
{
     SUnionMsgHeader    m_sHead;                   /* 8 */

    unsigned long      m_Tic;                      /* 4 bytes */
    unsigned long      ulSpare;                    /* 4 bytes */ unsigned
    short              awHalfWarns[CHANNELS_12]; /* 12*2 = 24 bytes */
                                                   /* each word is
                                                    * bit 0-2  L1 Half Cycle Warn
                                                    * bit 3 = L1 half cycle added
                                                    * bit 4-6  L2 Half Cycle Warn
                                                    * bit 7 = L2 half cycle added
                                                    * 8 =  LSB of 12 bit L1
SNR/4096
                                                    * 9 =  LSB of 12 bit L2
SNR/4096
                                                    * bit 10-15 Ktag of the SV */
    SMsg61Data         as61Data[CHANNELS_12];      /* 12*24 = 288 bytes */
    unsigned short     m_wCheckSum;                /* sum of all bytes of the
datalength */
    unsigned short     m_wCRLF;                    /* Carriage Return Line Feed */
} SBinaryMsg61;                                    /* length = 8 + (320) + 2 + 2 =
332 */
/***************************************************/
/*   SBinaryMsg66    GLONASS OBS (see notes on mesage 76) */
/***************************************************/
typedef struct
{
    SUnionMsgHeader    m_sHead;
    double             m_dTow;                      /* Time in seconds */
```

```
    unsigned short   m_wWeek;                    /* GPS Week Number */

    unsigned short   m_wSpare1;                  /* spare 1 (zero)*/

    unsigned long    m_ulSpare2;                 /* spare 2 (zero)*/

    SObsPacket       m_asL1Obs[CHANNELS_12];     /* 12 sets of L1(Glonass)
observations */

    SObsPacket       m_asL2Obs[CHANNELS_12];     /* 12 sets of L2(Glonass)
observations */

    unsigned long    m_aulL1CodeMSBsSlot[CHANNELS_12]; /* array of 12words.

                                                    bit 7:0 (8 bits) =
satellite Slot, 0

                                                    if no satellite

                                                    bit 12:8 (5 bits) = spare

                                                    bit 31:13 (19 bits) = upper
19 bits

                                                    of L1   LSB = 256 meters

                                                            MSB = 67108864
meters */

    unsigned short   m_wCheckSum;                /* sum of all bytes of the
datalength */

    unsigned short   m_wCRLF;                    /* Carriage Return Line Feed */

} SBinaryMsg66;                                  /* length = 8 + (352) + 2 + 2 =
364 */

/****************************************************/

/*   SGLONASS_String, added for glonass strings          */

/****************************************************/

typedef struct

{

   unsigned long m_aul85Bits[3];      /* holds bits 9-85 of the GLONASS string       */

                                      /*

                                       * bit order in message 65

                                       *                   MSB              LSB

                                       * m_aul85Bits[0]:  85 84.......... 54

                                       * m_aul85Bits[1]:  53 52.......... 22

                                       * m_aul85Bits[2]:  21 20..... 9

                                       */

} SGLONASS_String;                    /* 12 bytes (max of 96 bits) */
```

```
/**************************************************/

/*   SBinaryMsg65, added by JL for glonass subframe immediate data + string_5
 */

/**************************************************/

/* sent only upon command or when values change (not including changes in tk)
*/

typedef struct

{

    SUnionMsgHeader    m_sHead;

    unsigned char      m_bySV;                          /* The satellite to which
this data belongs. */

    unsigned char      m_byKtag;                        /* The satellite K Number + 8.
*/

    unsigned short     m_wSpare1;                       /* Spare, keeps alignment to
4 bytes */

    unsigned long      m_ulTimeReceivedInSeconds;       /* time at which this
arrived */

    SGLONASS_String    m_asStrings[5];                  /* first 5 Strings of
Glonass Frame (60 bytes) */

    unsigned short     m_wCheckSum;                     /* sum of all bytes of
datalength */

    unsigned short     m_wCRLF;                         /* Carriage Return Line
Feed */

} SBinaryMsg65;                                         /* length = 8 + (68) + 2 +
2 = 80 */

/*****************************************************************/

/*   SBinaryMsg62, Glonass almanac data. Containing string

 *   5 and the two string pair for each satellite after string 5.

 *   String 5 contains the time reference for the glonass almanac

 *   and gps-glonass time differences.

 *

 ****************************************************************/ typedef

struct

{

    SUnionMsgHeader    m_sHead;

    unsigned char      m_bySV;                          /* The satellite to which
this data belongs. */

    unsigned char      m_byKtag_ch;                     /* Proprietary data */
```

```
    unsigned short     m_wSpare1;                              /* Spare, keeps alignment to
4 bytes */

    SGLONASS_String    m_asStrings[3];                         /* glonass almanac data
 (36 bytes)

                                                                     0 & 1 = Two almanac SFs,
3= SF 5*/

    unsigned short     m_wCheckSum;                            /* sum of all bytes of the
datalength */

    unsigned short     m_wCRLF;                                /* Carriage Return Line
Feed */

} SBinaryMsg62;                                                /* length = 8 + (40) + 2 + 2
= 52 */

#if defined(WIN32) || (     ARMCC_VERSION >= 300441)

    #pragma pack(pop)

#endif

#ifdef    cplusplus

}

#endif

#endif //    BinaryMsg_H_
```

Current Version: v1.07 / February 16,
2017

Bin1 Message

Message Type [Binary]

Description GNSS position message (position and velocity data)

Command Format to Request Message

```
$JBIN,1,r<CR><LF>
```

where:

- '1' = Bin1 message
- **'r' = message rate in Hz (20, 10, 2, 1, 0, or .2)**

Message Format

Message Component	Description	Type	Bytes	Values
AgeOfDiff	Age of differential, seconds. Use Extended AgeOfDiff first. If both = 0, then no differential	Byte	1	0 to 255
NumOfSats	Number of satellites used in the GPS solution	Byte	1	0 to 12
GPSWeek	GPS week associated with this message	Unsigned short	2	0 to 65536
GPSTimeOfWeek	GPS tow (sec) associated with this message	Double	8	0.0 to 604800.0
Latitude	Latitude in degrees north	Double	8	-90.0 to 90.0
Longitude	Longitude in degrees East	Double	8	-180.0 to 180.0
Height	Altitude above the ellipsoid in meters	Float	4	
VNorth	Velocity north in m/s	Float	4	
VEast	Velocity east in m/s	Float	4	
Vup	Velocity up in m/s	Float	4	
StdDevResid	Standard deviation of residuals in meters	Float	4	Positive

NavMode	Navigation mode: 0 = No fix 1 = Fix 2d no diff 2 = Fix 3d no diff 3 = Fix 2D with diff 4 = Fix 3D with diff 5 = RTK float 6 = RTK integer fixed If bit 7 is set (left-most bit), then this is a manual position	Unsigned short	2	Bits 0 through 6 = Navmode Bit 7 = Manual mark
Extended AgeOfDiff	Extended age of differential, seconds. If 0, use 1 byte AgeOfDiff listed above	Unsigned short	2	0 to 65536

Structure

```
typedef struct

{

  SUnionMsgHeader  m_sHead;

  unsigned char    m_byAgeOfDiff;     /* age of differential, seconds
(255 max)*/

  unsigned char    m_byNumOfSats;     /* number of satellites used (12
max)    */

  unsigned short   m_wGPSWeek;        /* GPS week */

  double           m_dGPSTimeOfWeek;  /* GPS tow  */

  double           m_dLatitude;       /* Latitude degrees, -90..90 */

  double           m_dLongitude;      /* Longitude degrees, -180..180 */

  float            m_fHeight;         /* (m), Altitude ellipsoid */

  float            m_fVNorth;         /* Velocity north       m/s */

  float            m_fVEast;          /* Velocity eastm/s */

  float            m_fVUp;            /* Velocity up  m/s */

  float            m_fStdDevResid;    /* (m), Standard Deviation of
  Residuals */

  unsigned short   m_wNavMode;

  unsigned short   m_wAgeOfDiff;      /* age of diff using 16 bits   */

  unsigned short   m_wCheckSum;       /* sum of all bytes of the
datalength */
```

```
    unsigned short      m_wCRLF;            /* Carriage Return Line Feed */
  } SBinaryMsg1;                            /* length = 8 + 52 + 2 + 2 = 64 */
```

Additional Information	Message has a BlockID of 1 and is 52 bytes, excluding the header and epilogue
Related Commands	JBIN

Topic Last Updated: v1.07 / February 16, 2017

Bin2 Message

Message Type — Binary

Description

GPS DOPs (Dilution of Precision)

This message contains various quantities that are related to the GNSS solution, such as satellites tracked, satellites used, and DOPs.

Command Format to Request Message

```
$JBIN,2,r<CR><LF>
```

where:

- '2' = Bin2 message
- 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
MaskSatsTracked	Mask of satellites tracked by the GPS. Bit 0 corresponds to the GPS satellite with PRN 1.	Unsigned long	4	Individual bits represent satellites
MaskSatsUsed	Mask of satellites used in the GPS solution. Bit 0 corresponds to the GPS satellite with PRN 1.	Unsigned long	4	Individual bits represent satellites
GpsUtcDiff	Whole seconds between UTC and GPS time (GPS minus UTC)	Unsigned short	2	Positive
HDOPTimes10	Horizontal dilution of precision scaled by10 (0.1 units)	Unsigned short	2	Positive
VDOPTimes10	Vertical dilution of precision scaled by 10 (0.1 units)	Unsigned short	2	Positive
WAASMask	PRN and tracked or used status masks	Unsigned short	2	*See following*

- Bit 00 - Mask of satellites tracked by first WAAS satellite
- Bit 01 - Mask of satellites tracked by second WAAS satellite
- Bit 02 - Mask of satellites used by first WAAS satellite
- Bit 03 - Mask of satellites used by second WAAS satellite
- Bit 04 - Unused

- Bits 05-09 - Value used to find PRN of first WAAS satellite (This value + 120 = PRN)
- Bits 10-14 - Value used to find PRN of second WAAS satellite (This value + 120 = PRN)
- Bit 15 - Unused

Structure

```
typedef struct
{
  SUnionMsgHeader    m_sHead;
  unsigned long      m_ulMaskSatsTracked; /* SATS Tracked, bit mapped 0..31 */
  unsigned  long     m_ulMaskSatsUsed;    /* SATS Used, bit mapped 0..31 */

  unsigned short     m_wGpsUtcDiff;       /* GPS/UTC time difference (GPS minus
UTC) */

  unsigned short     m_wHDOPTimes10;      /* HDOP (0.1 units) */
  unsigned short     m_wVDOPTimes10;      /* VDOP (0.1 units) */
  unsigned short     m_wWAASMask;         /* Bits 0-1: tracked sats, Bits 2-3:
                                             used sats, Bits 5-9 WAAS PRN 1 minus
                                             120, Bits 10-14 WAAS PRN 1 minus 120
*/
  unsigned short     m_wCheckSum;         /* sum of all bytes of the datalength
*/
  unsigned short     m_wCRLF;             /* Carriage Return Line Feed */
} SBinaryMsg2;                            /* length = 8 + 16 + 2 + 2 = 28 */
```

Additional Information

Message has a BlockID of 2 and is 16 bytes, excluding the header and epilogue

Related Commands

JBIN

Topic Last Updated: v1.07 / February 16, 2017

Bin3 Message

Message Type Binary

Description Lat/Lon/Hgt, Covariances, RMS, DOPs and COG, Speed, Heading

Command Format to Request Message

`$JBIN,3,r<CR><LF>`

where:

- '3' = Bin3 message
- 'r' = message rate in Hz

Message Format

Message Component	Description	Type	Bytes	Values
GPSTimeOfWeek	GPS tow (sec) associated with this message	Double	8	0.0 to 604800.0
GPSWeek	GPS week associated with this message	Unsigned short	2	0 to 65536
SATS Tracked	Number of satellites tracked in the GPS solution	Unsigned short	2	
NumOfSats	Number of satellites used in the GPS solution	Byte	2	
NAV Mode	Navigation mode: 0 = No fix 1 = Fix 2d no diff 2 = Fix 3d no diff 3 = Fix 2D with diff 4 = Fix 3D with diff 5 = RTK float 6 = RTK integer fixed If bit 7 is set (left-most bit), then this is a manual position	unsigned char	1	Bits 0 through 6 = Navmode Bit 7 = Manual mark
Spare		unsigned char	1	
Latitude	Latitude in degrees north	Double	8	-90.0 to 90.0
Longitude	Longitude in degrees East	Double	8	-180.0 to 180.0
Height	Altitude above the ellipsoid in meters	Float	4	
Horizontal Speed	Velocity horizontal in m/s	Float	4	
Vup	Velocity up in m/s	Float	4	

COG	Course over Ground, degrees	Float	4	
Heading	Heading(degrees), Zero unless vecto	Float	4	
Pitch	Pitch (degrees), Zero unless vector	Float	4	
Spare		Float	4	
AgeOfDiff	Age of differential, seconds. Use Extended AgeOfDiff first. If both = 0, then no differential	Unsigned short	2	
Spare		Unsigned short	4	
Spare		Unsigned short	4	
Spare		Unsigned short	4	
HRMS	Horizontal RMS	Float	4	
VRMS	Vertical RMS	Float	4	
HDOP	Horizontal DOP	Float	4	
VDOP	Vertical DOP	Float	4	
TDOP	Time DOP	Float	4	
CovNN	Covaraince North-North	Float	4	
CovNE	Covaraince North-East	Float	4	
CovNU	Covaraince North-Up	Float	4	
CovEE	Covaraince East-East	Float	4	
CovEU	Covaraince East-Up	Float	4	
CovUU	Covaraince Up-Up	Float	4	

Structure typedef struct

```
{
    SUnionMsgHeader    m_sHead;                //
     Double            m_dGPSTimeOfWeek;          // GPS tow
    unsigned short     m_wGPSWeek;             // GPS week
    unsigned short     m_wNumSatsTracked;      // SATS Tracked
    unsigned short     m_wNumSatsUsed;      // SATS Used
    unsigned char      m_byNavMode;       // Nav Mode (same as message 1)
    unsigned char      m_bySpare00;       // Spare
    double             m_dLatitude;       // Latitude degrees, -90..90 double
                       m_dLongitude;      // Longitude degrees, -180..180
    float              m_fHeight;         // (m), Altitude ellipsoid float
                       m_fSpeed;          // Horizontal Speed     m/s
    float              m_fVUp;            // Vertical Velocity +up     m/s
    float              m_fCOG;            // Course over Ground, degrees
    float              m_fHeading;        // Heading (degrees), Zero unless vector
    float              m_fPitch;          // Pitch (degrees), Zero unless vector float
                       m_fSpare01;        // Spare
    unsigned short     m_wAgeOfDiff;      // age of differential, seconds
    unsigned short     m_wSpare02;        // Spare
    unsigned long      m_ulSpare03;       // Spare
    unsigned long      m_ulSpare04;       // Spare
    float              m_fHRMS;           // Horizontal RMS
    float              m_fVRMS;           // Vertical     RMS
    float              m_fHDOP;           // Horizontal DOP
    float              m_fVDOP;           // Vertical DOP
    float              m_fTDOP;           // Time DOP
    float              m_fCovNN;          // Covaraince North-North
    float              m_fCovNE;          // Covaraince North-East
    float              m_fCovNU;          // Covaraince North-Up
    float              m_fCovEE;          // Covaraince East-East
    float              m_fCovEU;          // Covaraince East-Up
    float              m_fCovUU;          // Covaraince Up-Up
    unsigned short     m_wCheckSum;       // sum of all bytes of the header and data
    unsigned short     m_wCRLF;           // Carriage Return Line Feed
} SBinaryMsg3;                            // length = 8 + 116 + 2 + 2 = 128      (108 =
74 hex)
```

Additional Information

Related Commands JBIN

Topic Last Updated: v1.07 / Octoter 13, 2016

Bin16 Message

Message Type	Binary
Description	Generic GNSS observations (see notes on message 76)
Command Format to Request Message	`$JBIN,16,r<CR><LF>` where: • '16' = Bin16 message • 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
Tow	Time in seconds	double	8	
Week	GPS week number	Unsigned short	2	Individual bits represent satellites
Spare1	Not used at this time	Unsigned short	2	Future Use
PageCount	Page information	Unsigned long	4	*See following*

- [0-15] Spare bits
- [16,17,18,19,20,21] Number of Pages = N
- [22,23,24,25,26,27] Page Number [0...N-1]
- [28,29,30,31] Spare bits

AllSignalsIncluded_01	Bit mask of all signals included in the set of pages	Unsigned long	4	*See following*

bit 0 = GPS:L1CA included
bit 1 = GPS:L2P included
bit 2 = GPS:L2C included
bit 3 = GPS:L5 included
bit 7:4 = spare
bit 8 = GLO:G1C or GLO:G1P included
bit 9 = GLO:G2C or GLO:G1P included
bit 15:10 = spare
bit 16 = GAL:E1BC included
bit 17 = GAL:E5A included
bit 18 = GAL:E5B included
bit 23:19 = spare
bit 24 = BDS:B1I included
bit 25 = BDS:B2I included
bit 26 = BDS:B3I included
bit 31:27 = spare

Message	Description	Type	Bytes	Values
AllSignalsInclude d_02	Bit mask of all signals included in the set of pages	Unsigned long	4	*See following*

- bit 0 = QZS:L1CA included
- bit 1 = spare
- bit 2 = QZS:L2C included
- bit 3 = QZS:L5 included
- bit 4 = QZS:L1C included
- bit 31:5 = spare

Obs[16]	16 sets of observations	Structure array	16*12 = 192	
CodeMSBsPRN	Array of 16 32-bit words	Array of unsigned longs	16*4=64	

- bit 7:0 (8 bits) = satellite PRN,
  = 0 if no satellite
- bit 12:8 (5 bits) = Log_Base_2(X+1)
  where X = Time, in units of 1/100th sec,
  since carrier phase tracking was last stressed
  or cycle slipped
- bit 31:13 (19 bits) = upper 19 bits
  of code pseudorange LSB = 256 meters
  MSB = 67108864 meters

ChanSignalSYS	Array of 16 16-bit words	Array of unsigned shorts	16*2=32	

- [15,14] spare bits
- [13] = 1 if GLONASS P-Code
- [12,11,10,9,8] = Channel (0 is the first channel)
- [7,6,5,4] = Signal ID (L1CA, L5, G1, B1I, B2I, B3I, etc)
  GPS Signal ID: L1CA=0, L2P=1, L2C=2, L5=3
  GLO Signal ID: G1C/G1P=0, G2C/G2P=1
  GAL Signal ID: E1BC=0, E5A=1, E5B=2
  BDS Signal ID: B1I=0, B2I=1, B3I=2
  QZS Signal ID: L1CA=0, L2C=2, L5=3, L1C=4
- [3,2,1,0] = GNSS System, 0=GPS,1=GLO,2=GAL,3=BDS,4=QZS

CheckSum	Sum of all bytes of header and data	Unsigned short	2	
CRLF	Carriage return line feed	Unsigned short	2	

Structure

```
typedef struct
```

```
{
    SUnionMsgHeader  m_sHead;                 //                 (8 bytes)
    double           m_dTow;                  // Time in seconds (8 bytes)
    unsigned short   m_wWeek;                 // GPS Week Number (2 bytes)
    unsigned short   m_wSpare1;               // spare 1 (zero)  (2 bytes)

    unsigned long    m_uPageCount; //[0-15] Spare bits
                               //[16,17,18,19,20,21] Number of Pages = N
                               //[22,23,24,25,26,27] Page Number [0...N-1]
                               //[28,29,30,31] Spare bits
                                          // Bit mask of all signals included
in the set of pages
    unsigned long    m_uAllSignalsIncluded_01;  // bit 0  = GPS:L1CA included
                                          // bit 1  = GPS:L2P included
                                          // bit 2  = GPS:L2C included
                                          // bit 3  = GPS:L5 included
                                          // bit 7:4 = spare
                                          // bit 8  = GLO:G1C or GLO:G1P
included
                                          // bit 9  = GLO:G2C or GLO:G1P
included
                                          // bit 15:10 = spare
                                          // bit 16 = GAL:E1BC included
                                          // bit 17 = GAL:E5A included
                                          // bit 18 = GAL:E5B included
                                          // bit 23:19 = spare
                                          // bit 24 = BDS:B1I included
                                          // bit 25 = BDS:B2I included
                                          // bit 26 = BDS:B3I included
                                          // bit 31:27 = spare
    unsigned long    m_uAllSignalsIncluded_02;  // bit 0  = QZS:L1CA included
                                          // bit 1  = spare
                                          // bit 2  = QZS:L2C included
                                          // bit 3  = QZS:L5 included
                                          // bit 4  = QZS:L1C included
                                          // bit 31:5 = spare
    SObsPacket       m_asObs[CHANNELS_gen];     // 16 sets of observations (16*12=192
bytes)
    unsigned long    m_aulCodeMSBsPRN[CHANNELS_gen]; // array of 16, 32 bit words
(16*4=64 bytes)
                                              // bit 7:0 (8 bits) = satellite
PRN,
                                              //                  = 0 if no
satellite
                                              // bit 12:8 (5 bits) =
Log_Base_2(X+1)
                                              //        where X = Time, in
units of 1/100th sec,
```

```
                                                        //        since carrier phase
tracking was last stressed
                                                        //        or cycle slipped
                                                        // bit 31:13 (19 bits) = upper 19
bits
                                                        // of code pseudorange LSB = 256
meters
                                                        //                    MSB =
67108864 meters
    unsigned short  m_awChanSignalSYS[CHANNELS_gen]; // Array of 16, 16 bit words (32
bytes)
                                                //[15,14]  spare bits
                                                //[13] = 1 if GLONASS P-Code
                                                //[12,11,10,9,8] = Channel (0 is the
first channel)
                                                //[7,6,5,4] = Signal ID (L1CA, L5,
G1, B1I, B2I, B3I, etc)
                                                // GPS Signal ID: L1CA=0, L2P=1,
L2C=2, L5=3
                                                // GLO Signal ID: G1C/G1P=0,
G2C/G2P=1
                                                // GAL Signal ID: E1BC=0, E5A=1,
E5B=2
                                                // BDS Signal ID: B1I=0, B2I=1, B3I=2
                                                // QZS Signal ID: L1CA=0, L2C=2,
L5=3, L1C=4
                                                //[3,2,1,0] = GNSS System,
0=GPS,1=GLO,2=GAL,3=BDS,4=QZS

    unsigned short   m_wCheckSum;               /// sum of all bytes of the header
and data (2 bytes)
    unsigned short   m_wCRLF;                   // Carriage Return Line Feed
(2 bytes)
} SBinaryMsg16;                                 // length = 8 +
(8+2+2+4+4+4+192+64+32=312) + 2 + 2 = 324
```

Additional Information

Related Commands JBIN

Topic Last Updated: v1.07 / February 16, 2017

Bin19 Message

Message Type	Binary
Description	GNSS diagnostic information
Command Format to Request Message	`$JBIN,19,r<CR><LF>` where: • '19' = Bin19 message • 'r' = message rate in Hz (1 or 0)

Message Component	Description	Type	Bytes	Values
SecOfWeek	Time of Week	long	4	
GPSWeek	GPS Week Number	unsigned short	2	
NavMode	Nav Mode	unsigned char	1	
UTCTimeDiff	Whole seconds between UTC and GPS time	char	1	
PageCount	Information about the paging of the BIN19 message.	unsigned long	4	Bits [16,17,18,19,20,21] Number of Pages = N Bits [22,23,24,25,26,27] Page Number [0...N-1]
AllSignalsIncludes01	Bitmask of all signals includes in this set of pages	unsigned long	4	bit 0 = GPS:L1CA included bit 1 = GPS:L2P included bit 2 = GPS:L2C included bit 3 = GPS:L5 included bit 8 = GLO:G1C or GLO:G1P included bit 9 = GLO:G2C or GLO:G1P included bit 16 = GAL:E1BC included bit 17 = GAL:E5A included bit 18 = GAL:E5B included bit 24 = BDS:B1I included bit 25 = BDS:B2I included bit 26 = BDS:B3I included
AllSignalsIncluded02	Continued bitmask of all signals included in this set of pages.	unsigned long	4	bit 0 = QZS:L1CA included bit 2 = QZS:L2C included bit 3 = QZS:L5 included bit 4 = QZS:L1C included
Spare		unsigned short	2	
ChannelData[16]	Detailed data for each signal included.	SGENERICchanData[]	320	
ChanSignalSYS	Information about the type of signal represented by each entry in ChannelData	unsigned short[]	32	[13] = 1 if GLONASS P-Code [12,11,10,9,8] = Channel (0 is the first channel) [7,6,5,4] = Signal ID (L1CA, L5, G1, B1I, B2I, B3I, etc) GPS Signal ID = 0: L1CA, 1: L2P, 2: L2C, 3: L5 GLO Signal ID = 0: G1C/G1P, 1: G2C/G2P GAL Signal ID = 0: E1BC, 1: E5A, 2:E5B BDS Signal ID = 0: B1I, 1: B2I, 2:B3I QZS Signal ID = 0: L1CA, 1: xxx, 2:L2C, 3: L5, 4: L1C [3,2,1,0] = GNSS System, 0=GPS,1=GLO,2=GAL,3=BDS,4=QZS
CheckSum	Sum of all bytes of header and data	Unsigned short	2	
CRLF	Carriage return line feed	Unsigned short	2	

Structure

```
//==================================================================
//  SGENERICchanData
//==================================================================
typedef struct
{
    unsigned char m_bySV;             // Bit (0-6) = SV slot, 0 == not tracked
    unsigned char m_byAlm_Ephm_Flags; // ephemeris and almanac status flags
                                      // bit 0: Ephemeris available but timed out
                                      // bit 1: Ephemeris valid
                                      // bit 2: Ephemeris health OK
                                      // bit 3: unused
                                      // bit 4: Almanac available
                                      // bit 5: Almanac health OK
                                      // bit 6: unused
                                      // bit 7: Satellite doesn't exist
    unsigned char m_byStatus;         // Status bits (code carrier bit frame...)
    char          m_chElev;           // elevation angle

    unsigned char m_byAzimuth;        // 1/2 the Azimuth angle
    unsigned char m_byLastMessage;    // last message processed
    unsigned char m_bySlip;           // cycle slip on chan 1
    char          m_cFlags;           //
                                      // [0] bChanEnabled
                                      // [1] bUsedInSolution

    unsigned short  m_wCliForSNR;     // code lock indicator for SNR divided by 32
    short           m_nDiffCorr;      // Differential correction * 100

    short           m_nDoppHz;        // expected doppler in HZ at B1 frequency
    short           m_nNCOHz;         // track from NCO in HZ

    short           m_nPosResid;      // position residual * 1000
    unsigned short  m_wAllocType;     //

} SGENERICchanData; // (20 bytes)

//--------------------------------------------------------------------------
//  SBinaryMsg19
//  Generic GNSS message for signal tracking status
//--------------------------------------------------------------------------
typedef struct
{
    SUnionMsgHeader   m_sHead;        // 8 bytes
    long              m_lSecOfWeek;   // tow (4 bytes)
    unsigned short    m_wGPSWeek;     // GPS Week Number (2 bytes)
    unsigned char     m_byNavMode;    // Nav Mode FIX_NO, FIX_2D, FIX_3D (high bit =has_diff)
```

```
    char              m_cUTCTimeDiff; // whole Seconds between UTC and GPS
    unsigned long     m_uPageCount;   // [0-15] Spare bits   (4 bytes)
                                      // [16,17,18,19,20,21] Number of Pages = N
                                      // [22,23,24,25,26,27] Page Number [0...N-1]
                                      // [28,29,30,31] Spare bits
                                                // Bit mask of all signals included in the set of pages
    unsigned long    m_uAllSignalsIncluded_01;  // bit 0  = GPS:L1CA included
                                                // bit 1  = GPS:L2P included
                                                // bit 2  = GPS:L2C included
                                                // bit 3  = GPS:L5 included
                                                // bit 7:4 = spare
                                                // bit 8  = GLO:G1C or GLO:G1P included
                                                // bit 9  = GLO:G2C or GLO:G1P included
                                                // bit 15:10 = spare
                                                // bit 16 = GAL:E1BC included
                                                // bit 17 = GAL:E5A included
                                                // bit 18 = GAL:E5B included
                                                // bit 23:19 = spare
                                                // bit 24 = BDS:B1I included
                                                // bit 25 = BDS:B2I included
                                                // bit 26 = BDS:B3I included
                                                // bit 31:27 = spare
    unsigned long    m_uAllSignalsIncluded_02;  // bit 0  = QZS:L1CA included
                                                // bit 1  = spare
                                                // bit 2  = QZS:L2C included
                                                // bit 3  = QZS:L5 included
                                                // bit 4  = QZS:L1C included
                                                // bit 31:5 = spare

    short             m_nClockErrAtL1;// clock error at L1, Hz  (2 bytes)
    unsigned short    m_wSpare1;      // spare (2 bytes)
    SGENERICchanData  m_asChannelData[CHANNELS_gen]; // channel data 16x20 = 320
    unsigned short  m_awChanSignalSYS[CHANNELS_gen]; // Array of 16, 16 bit words (32 bytes)
                                                //[15,14]  spare bits
                                                //[13] = 1 if GLONASS P-Code
                                                //[12,11,10,9,8] = Channel (0 is the first channel)
                                                //[7,6,5,4] = Signal ID (L1CA, L5, G1, B1I, B2I, B3I, etc)
                                                //   GPS Signal ID = 0: L1CA,     1: L2P,     2: L2C, 3: L5
                                                //   GLO Signal ID = 0: G1C/G1P,  1: G2C/G2P
                                                //   GAL Signal ID = 0: E1BC,     1: E5A,     2:E5B
                                                //   BDS Signal ID = 0: B1I,      1: B2I,     2:B3I
                                                //   QZS Signal ID = 0: L1CA,     1: xxx,     2:L2C,  3: L5,
4: L1C

                                                //[3,2,1,0] = GNSS System, 0=GPS,1=GLO,2=GAL,3=BDS,4=QZS

    unsigned short    m_wCheckSum;     // sum of all bytes of the header and data
    unsigned short    m_wCRLF;         // Carriage Return Line Feed
} SBinaryMsg19;                        // length = 8+(4+2+1+1+4+4+4+2+2+320+32)+2+2 = 8 + (376) + 2 + 2 =
```

Additional Information	
Related Commands	

Topic Last Updated: v1.07 / February 16, 2017

Bin35 Message

Message Type [Binary]

Description BeiDou ephemeris information

Command Format to Request Message

`$JBIN,35,r<CR><LF>`

where:

- '35' = Bin35 message
- **"r' = 1 (on) or 0 (off),**
  When set to on the message is sent once (one message for each tracked satellite at 1 second intervals) and then sent again whenever satellite information changes

Message Format

Message Component	Description	Type	Bytes	Values
SV	Satellite to which this data belongs	Unsigned short	2	
Spare1	Not used at this time	Unsigned short	2	Future use
SecOfWeek	Time at which this arrived (LSB=6)	Unsigned long	4	
BeiDouNav[30]	Unparsed BeiDou Navigation message	*See following*	4 x 30 = 120	

Elements correspond to the ephemeris values as defined in the BeiDou ICD:

1. Element 00, BDS_tow, Unsigned (4 bytes)
2. Element 01, BDS_toc, Unsigned (4 bytes)
3. Element 02, BDS_a0, Signed (4 bytes)
4. Element 03, BDS_a1,Signed (4 bytes)
5. Element 04, BDS_a2, Signed (4 bytes)
6. Element 05, BDS_toe, Unsigned (4 bytes)
7. Element 06, BDS_Root_A, Unsigned (4 bytes)
8. Element 07, BDS_Eccentricity, Unsigned (4 bytes)
9. Element 08, BDS_omega_perigee, Signed (4 bytes)
10. Element 09, BDS_DeltaN_MeanMotionDiff, Signed (4 bytes)
11. Element 10, BDS_M_MeanAnomaly, Signed (4 bytes)
12. Element 11, BDS_OMEGA0_Lon_Ascending, Signed (4 bytes)
13. Element 12, BDS_OMEGA_DOT, Signed (4 bytes)
14. Element 13, BDS_io_InclinationAngle, Signed (4 bytes)
15. Element 14, BDS_IDOT_RateInclination, Signed (4 bytes)
16. Element 15, BDS_Cuc_AmpCosHarmonicLat, Signed (4 bytes)

17. Element 16, BDS_Cus_AmpSinHarmonicLat, Signed (4 bytes)
18. Element 17, BDS_Crc_AmpCosHarmonicRadius, Signed (4 bytes)
19. Element 18, BDS_Crs_AmpSinHarmonicRadius, Signed (4 bytes)
20. Element 19, BDS_Cic_AmpCosHarmonicInclination, Signed (4 bytes)
21. Element 20, BDS_Cir_AmpSinHarmonicInclination, Signed (4 bytes)
22. Element 21, BDS_TGD1_TGD2, Unsigned (4 bytes)
    TGD1 in lower 10 bits (bits 0-9)
    TGD2 in next 10 bits (10-19)
23. Element 22, BDS_WN, Unsigned (4 bytes)
24. Element 23, BDS_alpha_0_1_2_3, Unsigned (4 bytes)
    Packed with 4, 8-bit words, exactly as defined in the BeiDou ICD Alpha3 in lower 8 bits (bits 0-7)
    Alpha2 in next 8 bits (bits 8-15)
    Alpha1 in next 8 bits (bits 16-23)
    Alpha0 in upper 8 bits (bits 24-31)
25. Element 24, BDS_beta_0_1_2_3, Unsigned (4 bytes)
    Packed with 4, 8-bit words, exactly as defined in the BeiDou ICD Beta3 in lower 8 bits (bits 0-7)
    Beta2 in next 8 bits (bits 8-15) Beta1 in next 8 bits (bits 16-23) Beta0 in upper 8 bits (bits 24-31)
26. Element 25, BDS_SatH1_IODC_URA1_IODE, Unsigned (4 bytes)
    IODE in lower 5 bits (bits 0-4)
    URA1 in next 4 bits (bits 5-8) IODC in next 5 bits (bits 9-13) SatH1in next 1 bit (bit 14)
27. Element 26, spare (4 bytes)
28. Element 27, spare (4 bytes)
29. Element 28, spare (4 bytes)
30. Element 29, spare (4 bytes)

Structure

```
typedef struct
{
  SUnionMsgHeader    m_sHead;
  unsigned short     m_wSV;             /* The satellite to which this data
belongs*/
  unsigned short     m_wSpare1;         /* spare 1 (chan number (as zero
9/1/2004) */
  unsigned long      m_TOW6SecOfWeek; /* time at which this arrived (LSB = 6sec)
*/
  unsigned long      m_BeidouNav[30]; /* Unparsed BeiDou navigation words.
  */
                                        /* Each of the BeiDou nav words contains one 32-
                                           bit signed or unsigned word.

Read
                                           as a signed or unsigned long as

needed. */

  unsigned short     m_wCheckSum;       /* sum of all bytes of the header and
data */
  unsigned short     m_wCRLF;           /* Carriage Return Line Feed */
} SBinaryMsg35;                         /* length = 8 + (128) + 2 + 2 = 140 */
```

Additional Information	**Message has a BlockID of 35 and is 128 bytes, excluding the header and epilogue**
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin36 Message

Message Type [Binary]

Description **BeiDou code and carrier phase information(all frequencies)**

Command Format to Request Message

`$JBIN,36,r<CR><LF>`

where:

- '36' = Bin36 message
- 'r' = message rate in Hz (20, 10, 2, 1, or 0)

Message Format

Message Component	Description	Type	Bytes	Values
Tow	Time in seconds	Double	2	
Week	GPS week number	Unsigned short	2	
Spare1	Spare 1 (zero)	Unsigned short	2	
FreqPage	See following	Unsigned long	4	
31. Bits 0-19 (20 bits) Spare bits 32. Bits 20-23 (4 bits) Number of pages 33. Bits 24-27 (4 bits) Page number 34. Bits 28-31 (4 bits) Signal ID (0 = B1I, 1 = B2I, 2 = B3I)				
Obs[CHANNELS_20]	20 sets of BeiDou observations	SObsPacket	20 x 12 = 240	
1CodeMSBsPRN[CHANNELS_20]	*See following*	Unsigned long	20 x 4 = 80	
• Bits 0-7 (8 bits) Satellite PRN, 0 if no satellite • Bits 8-12 (5 bits) Spare bits • Bits 13- 31 (19 bits) Upper 19 bits of B1/B2/B3, LSB = 256 meters, MSB = 67108864 meters				

Structure

```
typedef struct
{
```

```
  SUnionMsgHeader     m_sHead;                          //                   (8
bytes)
  double              m_dTow;                           // Time in seconds (8
bytes)
  unsigned short      m_wWeek;                          // GPS Week Number (2
bytes)
  unsigned short      m_wSpare1;                        // spare 1 (zero)    (2
bytes)

  unsigned long       m_uFreqPage;    //[0-19] Spare bits
                                      //[20,21,22,23] Number of Pages
                                      //[24,25,26,27] Page Number
                                      //[28,29,30,31] Signal ID (B1I, B2I, B3I,
etc)

      SObsPacket        m_asObs[CHANNELS_20];        // 20 sets of BeiDou observations
                                                        //    (20*12=240 bytes)
  unsigned long       m_aulCodeMSBsPRN[CHANNELS_20];    // array of 20 words
                                                          //    (20*4=80 bytes)
                                                          // bit 7:0 (8 bits) =
                                                          //    satellite PRN, 0
                                                          //    if no satellite
                                                          // bit 12:8 (5 bits) =
                                                          //    spare
                                                          // bit 31:13 (19 bits) =
                                                          //    upper 19 bits
                                                          //    of B1/B2/B3
                                                        //    LSB =       256
meters meters
                                                        //    MSB = 67108864

  unsigned short      m_wCheckSum;    // sum of all bytes of datalength
bytes)
  unsigned short      m_wCRLF;        // Carriage Return Line Feed          (2
} SBinaryMsg36;                       // length = 8 + (8+2+2+4+240+80) + 2 + 2 =
348
                                      //        = 8 + (336) + 2 + 2 = 348
```

Additional Information	
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin44 Message

Message Type	Binary
Description	Galileo time conversion parameters
Command Format to Request Message	`$JBIN,44,r<CR><LF>` where: • '44' = Bin44 message • 'r' = 1 (on) or 0 (off) When set to on the message is sent once and then sent again whenever satellite information changes

Message Format

Message Component	Description	Type	Bytes	Values
A0, A1	Coefficients for determining UTC time	Double	8 x 2 = 16	
tot	Reference time for A0 and A1, second of Galileo week	Unsigned long	4	
wnt	Current Galileo reference week	Unsigned short	2	
wnlsf	Week number when dtlsf becomes effective	Unsigned short	2	
dn	Day of week (1-7) when dtlsf becomes effective	Unsigned short	2	
dtls	Cumulative past leap seconds	Short	2	
dtlsf	Scheduled future leap seconds	Short	2	
Spare	Not used at this time	Short	2	Future use
A0G, A1G	Coefficients of GGTO polynomial	Double	8 x 2 = 16	
T0G	Reference time of week for GGTO	Unsigned long	4	
WN0G	Reference week for GGTO	Unsigned short	2	
GGTOisValid	Indicates if GGTO is valied	Unsigned short	2	0 = GGTO Invalid 1 = GGTO Valid.
CheckSum	Sum of all bytes of header and data	Unsigned short	2	
CRLF	Carriage return line feed	Unsigned short	2	

Structure

```
typedef struct
{
    // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    SUnionMsgHeader m_sHead;           // Header of message.
    // - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  (8 bytes)
    // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    // Galileo Time to UTC conversion parameters (32 bytes).
    double          m_A0;              // Constant term of polynomial to
                                       //  determine UTC from Galileo Time.
    double          m_A1;              // 1st order term of polynomial to
                                       //  determine UTC from Galileo Time.
    unsigned long   m_tot;             // Reference time for A0 & A1, sec of
                                       //  Galileo week.
    unsigned short  m_wnt;             // Current Galileo reference week.
    unsigned short  m_wnlsf;           // GST Week number when m_dtlsf
                                       //  becomes effective.
    unsigned short  m_dn;              // Day of the week 1 (= Sunday) to
```

```
                                        //  7 (= Saturday) when m_dtlsf
                                        //  becomes effective.
    short           m_dtls;             // Cumulative past leap seconds.
    short           m_dtlsf;            // Scheduled future (past) leap
                                        //  seconds.
    unsigned short  m_wSpare1;          // Spare (zero).
    // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - (32 bytes)
    // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    // GPS Time to Galileo Time conversion parameters (GGTO Parameters).
    //
    //   dTsys = Tgal - Tgps = m_A0G + m_A1G [TOW - m_t0G + 604800*(WN - m_WN0G)]
    //
    //     where,
    //        dTsys = The time difference between systems
    //        Tgal  = Galileo Time
    //        Tgps  = GPS Time
    //        TOW   = Galileo Time of Week
    //        WN    = Galileo Week Number
    //        remaining parameters follow.
    double          m_A0G;              // Constant term of GGTO polynomial.
    double          m_A1G;              // 1st order term of GGTO polynomial.
    unsigned long   m_t0G;              // Reference time of week for GGTO.
    unsigned short  m_WN0G;             // Reference week for GGTo.
    unsigned short  m_wGGTOisValid;     // Coded: 0 == GGTO Invalid,
                                        //        1 == GGTO Valid.
                                        //  The Galileo OS-SIS-ICD indicates
                                        //  that when satellite broadcasts
                                        //  all 1 bit values for A0G, A1G,
                                        //  t0G, and WN0G then "the GGTO is
                                        //  considered as not valid."
    // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - (24 bytes)
    // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    // Message Tail
    unsigned short  m_wCheckSum;        // Sum of all bytes of the header and
                                        //  data.
    unsigned short  m_wCRLF;            // Carriage Return Line Feed.
    // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  (4 bytes)
} SBinaryMsg44;                     // length = 8 + (32+24) + 2 + 2 = 68.
```

Additional Information Message has a BlockID of 44 and is 56 bytes, excluding the header and epilogue

Related Commands JBIN

Topic Last Updated: v1.07 / February 16, 2017

Bin45 Message

Message Type [Binary]

Description Galileo ephemeris information

Command Format to Request Message

$JBIN,45,r<CR><LF>

where:

- '45' = Bin45 message
- **"r' = 1 (on) or 0 (off),**
  When set to on the message is sent once (one message for each tracked satellite at 1 second intervals) and then sent again whenever satellite information changes

Message Format

Message Component	Description	Type	Bytes	Values
SV	Satellite to which this data belongs	Unsigned short	2	
Spare1	Not used at this time	Unsigned short	2	Future use
SecOfWeek	Time at which this arrived (LSB = 6)	Unsigned long	4	
SF1words[10]	Unparsed SF 1 message	Unsigned long	4 x 10 = 40	
SF2words[10]	Unparsed SF 2 message	Unsigned long	4 x 10 = 40	
SF3words[10]	Unparsed SF 3 message	Unsigned long	4 x 10 = 40	

Structure

```
typedef struct
            {
              SUnionMsgHeader  m_sHead;
              unsigned short   m_wSV;               /* The satellite to which this data belongs.
            *
            /
              unsigned short     m_wSpare1;         /* spare 1 (chan number (as zero 9/1/2004)*/
              unsigned long      m_TOW6SecOfWeek; /* time at which this arrived (LSB =6sec)
            *
            /
              unsigned long m_SF1words[10]; /* Unparsed SF 1 message words. */ unsigned
              long m_SF2words[10]; /* Unparsed SF 2 message words. */ unsigned long
              m_SF3words[10]; /* Unparsed SF 3 message words.*/
```

```
                                    /* Each of the subframe words contains
                                       one 30-bit GPS word in the lower
                                       30 bits, The upper two bits are ignored
                                       Bits are placed in the words from left to
                                       right as they are received */
   unsigned short   m_wCheckSum;    /* sum of all bytes of the datalength */
   unsigned short   m_wCRLF;        /* Carriage Return Line Feed */
 } SBinaryMsg95;                    /* length = 8 + (128) + 2 + 2 = 140 */
```

Additional Information

Message has a BlockID of 45 and is 128 bytes, excluding the header and epilogue

Related Commands JBIN

Topic Last Updated: v1.07 / February 16, 2017

Bin62 Message

Message Type Binary, GLONASS

Description GLONASS almanac information

Command Format to Request Message

```
$JBIN,62,r<CR><LF>
```

where:

- '62' = Bin62 message
- 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
SV	Satellite to which this data belongs	Byte	1	
Ktag_ch	Proprietary data	Byte	1	
Spare1	Spare, keeps alignment to 4 bytes	Unsigned short	2	
Strings[3]	GLONASS almanac data (36 bytes) • 0 & 1 = Two almanac SFs • 3= SF 5	SGLONASS string	36	

Structure

```
typedef struct
{
  SUnionMsgHeader     m_sHead;
  unsigned char       m_bySV;          /* The satellite to which this data
belongs. */
  unsigned char       m_byKtag_ch;     /* Proprietary data */
  unsigned short      m_wSpare1;       /* Spare, keeps alignment to 4 bytes */
  SGLONASS_String     m_asStrings[3]; /* glonass almanac data        (36 bytes)
                                           0 & 1 = Two almanac SFs, 3= SF 5*/
  unsigned short      m_wCheckSum;     /* sum of all bytes of the datalength */
  unsigned short      m_wCRLF;         /* Carriage Return Line Feed */
    } SBinaryMsg62;                        /* length = 8 + (40) + 2 + 2 = 52 */
```

Additional Information	
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin65 Message

Message Type Binary, GLONASS

Description GLONASS ephemeris information

Command Format to Request Message

```
$JBIN,65,r<CR><LF>
```

where:

- '65' = Bin65 message
- **"r' = 1 (on) or 0 (off),**
  When set to on the message is sent once (one message for each tracked satellite at 1 second intervals) and then sent again whenever satellite information changes

Message Format

Message Component	Description	Type	Bytes	Values
SV	Satellite to which this data belongs	Byte	1	
Ktag	Satellite K Number + 8	Byte	1	
Spare1	Spare, keeps alignment to 4 bytes	Unsigned short	2	
TimeReceivedInSeconds	Time at which this arrived	Unsigned long	4	
Strings[5]	First five strings of GLONASS frame (60 bytes)	SGLONASS string	60	

Structure

```
typedef struct

{

  SUnionMsgHeader m_sHead;

  unsigned char     m_bySV;            /* The satellite to which this data
belongs. */

  unsigned char     m_byKtag;          /* The satellite K Number + 8. */ unsigned

  short             m_wSpare1;         /* Spare, keeps alignment to 4 bytes */

  unsigned long     m_ulTimeReceivedInSeconds; /* time at which this arrived*/

  SGLONASS_String m_asStrings[5]; /* first 5 Strings of Glonass Frame (60 bytes)
*/

  unsigned short  m_wCheckSum;     /* sum of all bytes of the datalength */

  unsigned short  m_wCRLF;         /* Carriage Return Line Feed */
```

```
} SBinaryMsg65;                    /* length = 8 + (68) + 2 + 2 = 80 */
```

Additional Information	
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin66 Message

Message Type: Binary, GLONASS

Description: GLONASS L1/L2 code and carrier phase information

---

Command Format to Request Message

`$JBIN,66,r<CR><LF>`

where:

- '66' = Bin66 message
- 'r' = message rate in Hz (20, 10, 2, 1, or 0)

Message Format

Message Component	Description	Type	Bytes	Values
Tow	Time in seconds	Double		
Week	GPS week number	Unsigned short		
Spare1	Spare 1 (zero)	Unsigned short		
Spare2	Spare 2 (zero)	Unsigned long		
L1Obs[CHANNELS_12]	12 sets of L1 (GLONASS) observations	SObsPacket		
L2Obs[CHANNELS_12]	12 sets of L2 (GLONASS) observations	SObsPacket		
L1CodeMSBsSlot[CHANNELS_12]	*See following*	Unsigned long		

- Bits 0-7 (8 bits)
  Satellite slot, 0 if no satellite
- Bits 8-12 (5 bits)
  Spare bit
- Bits 13- 31 (19 bits)
  Upper 19 bits of L1, LSB = 256 meters, MSB = 67108864 meters

---

Structure

```
typedef struct
{
  SUnionMsgHeader m_sHead;
  double          m_dTow;          /* Time in seconds */
  unsigned short  m_wWeek;         /* GPS Week Number */
```

```
    unsigned short     m_wSpare1;         /* spare 1 (zero)*/
    unsigned long      m_ulSpare2;        /* spare 2 (zero)*/
    SObsPacket         m_asL1Obs[CHANNELS_12];     /* 12 sets of L1(Glonass)
                                                      observations */
    SObsPacket         m_asL2Obs[CHANNELS_12];     /* 12 sets of L2(Glonass)
                                                      observations */
    unsigned long      m_aulL1CodeMSBsSlot[CHANNELS_12]; /* array of 12 words.
                                                      bit 7:0 (8 bits) =
                                                      satellite Slot, 0 if
no
                                                      satellite
                                                      bit 12:8 (5 bits) =
  spare
                                                      bit 31:13 (19 bits) =
                                                      upper 19 bits of L1 LSB
                                                      = 256 meters
                                           MSB = 67108864 meters
*/
    unsigned short     m_wCheckSum;       /* sum of all bytes of the datalength */
    unsigned short     m_wCRLF;           /* Carriage Return Line Feed */
  } SBinaryMsg66;                         /* length = 8 + (352) + 2 + 2 = 364 */
```

Additional Information	
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin69 Message

Message Type: Binary, GLONASS

Description: GLONASS L1/L2diagnostic information

Command Format to Request Message

```
$JBIN,69,r<CR><LF>
```

where:

- '69' = Bin69 message
- 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
SecOfWeek	Tow	Long		
L1usedNavMask	Mask of L1 channels used in nav solution	Unsigned short		
L2usedNavMask	Mask of L2 channels used in nav solution	Unsigned short		
ChannelData[CHANNELS_12]	Channel data 12X24 = 288	SGLONASSChanData		
Week	Week	Unsigned short		
Spare01	Spare 1	Unsigned char		
Spare02	Spare 2	Unsigned char		

Structure

```
typedef struct
{
  SUnionMsgHeader m_sHead;
  long             m_lSecOfWeek;       /* tow   */
  unsigned short   m_wL1usedNavMask; /* mask of L1 channels used in nav
solution */
  unsigned short   m_wL2usedNavMask; /* mask of L2 channels used in nav
solution */
  SGLONASSChanData    m_asChannelData[CHANNELS_12]; /* channel data 12X24 = 288
*/
```

```
  unsigned short    m_wWeek;             /* week */
  unsigned char     m_bySpare01;         /* spare 1 */
  unsigned char     m_bySpare02;         /* spare 2 */
  unsigned short    m_wCheckSum;         /* sum of all bytes of the datalength */
  unsigned short    m_wCRLF;             /* Carriage Return Line Feed */
} SBinaryMsg69;                          /* length = 8 + 300 + 2 + 2 = 312 */
```

Additional Information	
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin76 Message

Message Type Binary

Description

GPS L1/L2 code and carrier phase information

Note: "Code" means pseudorange derived from code phase. "Phase" means range derived from carrier phase. This will contain cycle ambiguities.

Only the lower 16 bits of L1P code, L2P code and the lower 23 bits of carrier phase are provided. The upper 19 bits of the L1CA code are found in m_aulCACodeMSBsPRN[]. The upper 19 bits of L1P or L2P must be derived using the fact L1P and L2P are within 128 m (419.9 ft) of L1CA.

To determine L1P or L2P:

1. Use the lower 16 bits provided in the message.
2. Set the upper bits to that of L1CA.
3. Add or subtract on LSB of the upper bits (256 meters (839.9 feet)) so that L1P or L2P are with in 1/2 LSB (128 m (419.9 ft))

The carrier phase is in units of cycles, rather than meters, and is held to within 1023 cycles of the respective code range. Only the lower 16+7 = 23 bits of carrier phase are transmitted in Bin 76.

To determine the remaining bits:

1. Convert the respective code range (determined above) into cycles by dividing by the carrier wavelength. This is the nominal reference phase.
2. Extract the 16 and 7 bit blocks of carrier phase from bin 76 and arrange it to form the lower 23 bits of carrier phase.
3. Set the upper bits (bit 23 and above) equal to those of the nominal reference phase
4. Add or subtract the least significant upper bit (8192 cycles) so that carrier phase most closely agrees with the nominal reference phase (to within 4096 cycles).

Command Format to Request Message

```
$JBIN,76,r<CR><LF>
```

where:

- '76' = Bin76 message
- **'r' = message rate in Hz (20, 10, 2, 1, 0, or .2)**

Message Format

Message Component	Description	Type	Bytes	Values
TOW	Predicted GPS time in seconds	Double	8	
Week	GPS week number	Unsigned short	2	
Spare1		Unsigned long	2	
Spare2		Unsigned long	4	
L2PSatObs[12] *(array for next 3 fields)*	*L2 satellite observation data*	*Structure array*	*12 x 12 = 144*	
CS_TT_W3_SNR	*See following*	Unsigned long	4	

- Bits 0-11 (12 bits)
  SNR; 10.0 X log10(0.1164xSNR_value)
- Bits 12-14 (3 bits)
  Cycle Slip Warn (warning for potential 1/2 cycle slips); a warning exists if any of these bits are

set • Bit 15 (1 bit) Long Track Time;1 if Track Time > 25.5 sec (0 otherwise) • Bits 16-23 (8 bits) Track Time (signal tracking time in seconds); LSB = 0.1 seconds; Range = 0 to 25.5 seconds • Bits 24-31 (8 bits) Cycle Slips; increments by 1 every cycle slip with natural roll-over after 255				
P7_Doppler_FL	*See following*	Unsigned long	4	
• Bit 0 (1 bit) Phase Valid (Boolean);1 if valid phase (0 otherwise) • Bits 1-23 (23 bits) Doppler (magnitude of Doppler);LSB = 1/512 cycle/sec; Range = 0 to 16384 cycle/sec • Bit 24 (1 bit) Doppler Sign (sigh of Doppler);1 = negative, 0 = positive • Bits 25-31 (7 bits) Carrier Phase (High part) (Upper 7 bits of the 23 bit carrier phase): LSB = 64 cycles, MSB = 4096 cycles				
CodeAndPhase	*See following*	Unsigned long	4	
• Bits 0-15 (16 bits) Pseudorange (lower 16 bits of code pseudorange);LSB = 1/256 meters, MSB = 128 meters **Note:** For CA code, the upper 19 bits are given in L1CACodeMSBsPRN[] below • Bits 16-31 (16 bits) Carrier Phase (lower 16 bits of the carrier phase); LSB = 1/1024 cycles, MSB = 32 cycles **Note:** The 7 MSBs are given in P7_Doppler_FL (see preceding row in this table)				
L1CASatObs[15] (array for next 3 fields)	*L1 satellite code observation data*	*Structure array*	*15 x 12 = 180*	
CS_TT_W3_SNR	*See following*	Unsigned long	4	
• Bits 0-11 (12 bits) SNR; 10.0 X log10(0.1024xSNR_value) • Bits 12-14 (3 bits) Cycle Slip Warn (warning for potential 1/2 cycle slips); a warning exists if any of these bits are set • Bit 15 (1 bit) Long Track Time;1 if Track Time > 25.5 sec (0 otherwise) • Bits 16-23 (8 bits) Track Time (signal tracking time in seconds); LSB = 0.1 seconds; Range = 0 to 25.5 seconds • Bits 24-31 (8 bits) Cycle Slips; increments by 1 every cycle slip with natural roll-over after 255				

<table>
<tr><td>P7_Doppler_FL</td><td>See following</td><td>Unsigned long</td><td>4</td><td></td></tr>
<tr><td colspan="5">

Bit 0 (1 bit)
Phase Valid (Boolean);1 if valid phase (0 otherwise)
Bits 1-23 (23 bits)
Doppler (magnitude of Doppler);LSB = 1/512 cycle/sec; Range = 0 to 16384 cycle/sec
Bit 24 (1 bit)
Doppler Sign (sigh of Doppler);1 = negative, 0 = positive
Bits 25-31 (7 bits)
Carrier Phase (High part) (Upper 7 bits of the 23 bit carrier phase): LSB = 64 cycles, MSB = 4096 cycles

</td></tr>
<tr><td>CodeAndPhase</td><td>See following</td><td>Unsigned long</td><td>4</td><td></td></tr>
<tr><td colspan="5">

Bits 0-15 (16 bits)
Pseudorange (lower 16 bits of code pseudorange);LSB = 1/256 meters, MSB = 128 meters
Note: For CA code, the upper 19 bits are given in L1CACodeMSBsPRN[] below
Bits 16-31 (16 bits)
Carrier Phase (lower 16 bits of the carrier phase); LSB = 1/1024 cycles, MSB = 32 cycles
Note: The 7 MSBs are given in P7_Doppler_FL (see preceding row in this table)

</td></tr>
<tr><td>L1CACodeMSBsPRN[15]</td><td>L1CA code observation</td><td>Array of 15 Unsigned long</td><td>15 x 4 = 60</td><td>See following</td></tr>
<tr><td colspan="5">

Bits 0-7 (8 bits)
PRN (space vehicle ID);PRN = 0 if no data
Bits 8-12 (5 bits)
Unused
Bits 13-31 (19 bits)
L1CA Range (upper 19 bits of L1CA); LSB = 256 meters, MSB = 67,108,864 meters

</td></tr>
<tr><td>L1PCode[12]</td><td>L1(P) code observation data</td><td>Array of 12 Unsigned long</td><td>12 x 4 = 48</td><td>See following</td></tr>
<tr><td colspan="5">

Bits 0-15 (16 bits)
L1P Range (lower 16 bits of the L1P code pseudorange);LSB = 1/256 meters, MSB = 128 meters
Bits 16-27 (12 bits)
L1P SNR (L1P signal-to-noise ratio); SNR = 10.0 x log(0.1164 x SNR_value), if 0, then L1P channel not tracked
Bits 28-31 (4 bits)
Unused

</td></tr>
<tr><td>wCeckSum</td><td>Sum of all bytes of header and data</td><td>Unsigned short</td><td>2</td><td></td></tr>
<tr><td>wCRLF</td><td>Carriage return line feed</td><td>Unsigned short</td><td>2</td><td></td></tr>
</table>

Structure

```
typedef struct
{
  SUnionMsgHeader m_sHead;
  double           m_dTow;             /* GPS Time in seconds */
  unsigned short   m_wWeek;            /* GPS Week Number */
  unsigned short   m_wSpare1;          /* spare 1 (zero)*/ unsigned
  long             m_ulSpare2;         /* spare 2 (zero)*/
  SObsPacket       m_asL2PObs[CHANNELS_12];        /* 12 sets of L2(P) observations
*/
  SObsPacket       m_asL1CAObs[CHANNELS_L1_E];     /* 15 sets of L1(CA) observations
*/
  unsigned long    m_aulCACodeMSBsPRN[CHANNELS_L1_E]; /* array of 15words.
                                                          bit 7:0 (8 bits) =
satellite
                                                          PRN, 0 if no satellite
                                                            bit 12:8 (5 bits) =
spare
                                                          bit 31:13 (19 bits) =
 upper
                                                          19 bits of L1CA LSB
                                                          = 256 meters
                                                          MSB = 67108864 meters */
  unsigned long    m_auL1Pword[CHANNELS_12];     /* array of 12 words relating to
L1(P)
                                                 code. Bit 0-15 (16 bits) lower bits of
 16
                                                 the L1P code pseudo
 range.
                                                 LSB = 1/256 meters MSB
                                                 = 128 meters
                                                 Bits 16-27 (12 bits) = L1P
SNR_value
                                                 SNR = 10.0*log10(
0.1164*SNR_value)
                                                 If Bits 16-27 all zero, no L1P
track
```

```
                                        Bits 28-31 (4 bits) spare */
  unsigned short  m_wCheckSum;        /* sum of all bytes of the datalength */
  unsigned short  m_wCRLF;            /* Carriage Return Line Feed */
} SBinaryMsg76;                       /* length = 8 + (448) + 2 + 2 = 460 */
```

Additional Information	
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin80 Message

Message Type Binary

Description SBAS data frame information

Command Format to Request Message

`$JBIN,80,r<CR><LF>`

where:

- '80' = Bin80 message
- 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
PRN	Broadcast PRN	Unsigned short	2	
Spare	Not used at this time	Unsigned short	2	Future use
MsgSecOfWeek	Seconds of week for message	Unsigned long	4	
WaasMsg[8]	250-bit WAAS message (RTCA DO0229). 8 unsigned longs, with most significant bit received first.	Unsigned long	4 x 8 = 32	

Structure

```
typedef struct
{
  SUnionMsgHeader    m_sHead;
  unsigned short m_wPRN;                 /* Broadcast PRN */
  unsigned short m_wSpare;               /* spare (zero) */
  unsigned long     m_ulMsgSecOfWeek;    /* Seconds of Week For Message */ unsigned
  long              m_aulWaasMsg[8];     /* Actual 250 bit waas message*/ unsigned
  short m_wCheckSum;                     /* sum of all bytes of the datalength */
  unsigned short m_wCRLF;                /* Carriage Return Line Feed */
} SBinaryMsg80;                          /* length = 8 + (40) + 2 + 2 = 52 */
```

Additional Information Message has a BlockID of 80 and is 40 bytes, excluding the header and epilogue

Related JBIN

Commands

Topic Last Updated: v1.06 / March 10, 2015

Bin89 Message

Message Type: Binary

Description: SBAS satellite tracking information (supports three SBAS satellites)

Command Format to Request Message

```
$JBIN,89,r<CR><LF>
```

where:

- '89' = Bin89 message
- 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
GPSSecOfWeek	GPS tow integer sec	Long		
MaskSBASTracked	SBAS satellites tracked, bit mapped 0..3	Byte		
MaskSBASUSED	SBAS satellites used, bit mapped 0..3	Byte		
Spare	Spare	Unsigned short		
ChannelData[CHANNELS_SBAS_E]	SBAS channel data	SChannelData		

Structure

```
typedef struct
{
  SUnionMsgHeader    m_sHead;
  long               m_lGPSSecOfWeek;      /* GPS tow integer sec */
  unsigned char      m_byMaskSBASTracked; /* SBAS Sats Tracked, bit mapped 0..3*/
  unsigned char      m_byMaskSBASUSED;     /* SBAS Sats Used, bit mapped 0..3 */
  unsigned short     m_wSpare;             /* spare */

  SChannelData       m_asChannelData[CHANNELS_SBAS_E];      /* SBAS channel data */
  unsigned short     m_wCheckSum;          /* sum of all bytes of the datalength */
  unsigned short     m_wCRLF;              /* Carriage Return Line Feed */
} SBinaryMsg89;                            /* length = 8 + 80 + 2 + 2 = 92 */
```

Additional Information	
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin93 Message

Message Type [Binary]

Description SBAS ephemeris information

Command Format to Request Message

```
$JBIN,93,r<CR><LF>
```

where:

- '93' = Bin93 message
- 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
SV	Satellite to which this data belongs	Unsigned short	2	
Spare	Not used at this time	Unsigned short	2	Future use
TOWSecOfWeek	Time at which this arrived (LSB = 1 sec)	Unsigned long	4	
IODE		Unsigned short	2	
URA	Consult the [ICD-GPS-200] for definition in Appendix A	Unsigned short	2	
TO	Bit 0 = 1 sec	Long	4	
XG	Bit 0 = 0.08 m	Long	4	
YG	Bit 0 = 0.08 m	Long	4	
ZG	Bit 0 = 0.4 m	Long	4	
XGDot	Bit 0 = 0.000625 m/sec	Long	4	
YXDot	Bit 0 = 0.000625 m/sec	Long	4	
ZGDot	Bit 0 = 0.004 m/sec	Long	4	
XGDotDot	Bit 0 = 0.0000125 m/sec/sec	Long	4	
YGDotDot	Bit 0 = 0.0000125 m/sec/sec	Long	4	
ZGDotDot	Bit 0 = 0.0000625 m/sec/sec	Long	4	
Gf0	Bit 0 = 2**-31 sec	Unsigned short	2	
Gf0Dot	Bit 0 = 2**-40sec/sec	Unsigned short	2	

Structure

```
typedef struct
{
  SUnionMsgHeader m_sHead;
```

```
  unsigned short  m_wSV;      /* The satellite to which this data belongs. */
  unsigned short  m_wWeek;    /* Week corresponding to m_lTOW*/
  unsigned long   m_lSecOfWeekArrived; /* time at which this arrived (LSB = 1sec)
*/
  unsigned short  m_wIODE;
  unsigned short  m_wURA;     /* See 2.5.3 of Global Pos Sys Std Pos Service Spec
*/
  long    m_lTOW;             /* Sec of WEEK Bit 0 = 1 sec */
  long    m_lXG;              /* Bit 0 = 0.08 m */
  long    m_lYG;              /* Bit 0 = 0.08 m */
  long    m_lZG;              /* Bit 0 = 0.4 m */
  long    m_lXGDot;           /* Bit 0 = 0.000625 m/sec */
  long    m_lYGDot;           /* Bit 0 = 0.000625 m/sec */
  long    m_lZGDot;           /* Bit 0 = 0.004 m/sec */
  long    m_lXGDotDot;      /* Bit 0 = 0.0000125 m/sec/sec */ long
  m_lYGDotDot;    /* Bit 0 = 0.0000125 m/sec/sec */ long  m_lZGDotDot;
  /* Bit 0 = 0.0000625 m/sec/sec */  short  m_nGf0; /* Bit 0 = 2**-31
  sec */
  short  m_nGf0Dot;         /* Bit 0 = 2**-40 sec/sec */
  unsigned short  m_wCheckSum; /* sum of all bytes of the datalength */
  unsigned short  m_wCRLF;    /* Carriage Return Line Feed */
} SBinaryMsg93;               /* length = 8 + (56) + 2 + 2 = 68 */
```

Additional Information	Message has a BlockID of 93 and is 45 bytes, excluding the header and epilogue
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin94 Message

Message Type [Binary]

Description Ionospheric and UTC conversion parameters

Command Format to Request Message

```
$JBIN,94,r<CR><LF>
```

where:

- '94' = Bin94 message
- 'r' = 1 (on) or 0 (off)
  When set to on the message is sent once and then sent again whenever satellite information changes

Message Format

Message Component	Description	Type	Bytes	Values
a0, a1,a2, a3	AFCRL alpha parameters	Double	8 x 4 = 32	
b0, b1,b2, b3	AFCRL beta parameters	Double	8 x 4 = 32	
A0, A1	Coefficients for determining UTC time	Double	8 x 2 = 16	
tot	Reference time for A0 and A1, second of GPS week	Unsigned long	4	
wnt	Current UTC reference week	Unsigned short	2	
wnlsf	Week number when dtlsf becomes effective	Unsigned short	2	
dn	Day of week (1-7) when dtlsf becomes effective	Unsigned short	2	
dtls	Cumulative past leap	Short	2	
dtlsf	Scheduled future leap	Short	2	
Spare	Not used at this time	Short	2	Future use

Structure

```
typedef struct
{
  SUnionMsgHeader    m_sHead;
  /* Iono parameters. */
  double           m_a0,m_a1,m_a2,m_a3;     /* AFCRL alpha parameters. */
  double           m_b0,m_b1,m_b2,m_b3;     /* AFCRL beta parameters.     */
  /* UTC conversion parameters. */
```

```
  double          m_A0,m_A1;     /* Coeffs for determining UTC time. */
  unsigned long   m_tot;         /* Reference time for A0 & A1, sec of GPS week. */
  unsigned short m_wnt;          /* Current UTC reference week number. */
  unsigned short m_wnlsf;        /* Week number when dtlsf becomes effective. */
  unsigned short m_dn;           /* Day of week (1-7) when dtlsf becomes effective.
*/
  short           m_dtls;        /* Cumulative past leap seconds. */
  short           m_dtlsf;       /* Scheduled future leap seconds. */
  unsigned short m_wSpare1;      /* spare 4 (zero)*/
  unsigned short m_wCheckSum; /* sum of all bytes of the datalength*/
  unsigned short m_wCRLF;        /* Carriage Return Line Feed */
} SBinaryMsg94;                  /* length = 8 + (96) + 2 + 2 =      108 */
```

Additional Information	Message has a BlockID of 94 and is 96 bytes, excluding the header and epilogue
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin95 Message

Message Type [Binary]

Description GPS ephemeris information

Command Format to Request Message

$JBIN,95,r<CR><LF>

where:

- '95' = Bin95 message
- 'r' = 1 (on) or 0 (off)
When set to on the message is sent once (one message for each tracked satellite at 1 second intervals) and then sent again whenever satellite information changes

Message Format

Message Component	Description	Type	Bytes	Values
SV	Satellite to which this data belongs	Unsigned short	2	
Spare1	Not used at this time	Unsigned short	2	Future use
SecOfWeek	Time at which this arrived (LSB = 6)	Unsigned long	4	
SF1words[10]	Unparsed SF 1 message	Unsigned long	4 x 10 = 40	
SF2words[10]	Unparsed SF 2 message	Unsigned long	4 x 10 = 40	
SF3words[10]	Unparsed SF 3 message	Unsigned long	4 x 10 = 40	

Structure

```
typedef struct

{

  SUnionMsgHeader  m_sHead;

  unsigned short   m_wSV;               /* The satellite to which this data belongs.
*/

  unsigned short     m_wSpare1;         /* spare 1 (chan number (as zero 9/1/2004)*/

  unsigned long      m_TOW6SecOfWeek; /* time at which this arrived (LSB =6sec)
*/

  unsigned long m_SF1words[10]; /* Unparsed SF 1 message words. */ unsigned long

  m_SF2words[10];  /*  Unparsed  SF  2  message  words.  */  unsigned  long

  m_SF3words[10];  /* Unparsed SF 3 message words.*/

                                        /* Each of the subframe words contains
```

```
                                        one 30-bit GPS word in the lower
                                        30 bits, The upper two bits are ignored
                                        Bits are placed in the words from left to
                                        right as they are received */
  unsigned short   m_wCheckSum;      /* sum of all bytes of the datalength */
  unsigned short   m_wCRLF;          /* Carriage Return Line Feed */
} SBinaryMsg95;                      /* length = 8 + (128) + 2 + 2 = 140 */
```

Additional Information	Message has a BlockID of 95 and is 128 bytes, excluding the header and epilogue
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin96 Message

Message Type [Binary](#)

Description GPS L1 code and carrier phase information

Command Format to Request Message

`$JBIN,96,r<CR><LF>`

where:

- '96' = Bin96 message
- 'r' = message rate in Hz (20, 10, 2, 1, or 0)

Message Format

Message Component	Description	Type	Bytes	Values
Spare1	Not used at this time	Unsigned short	2	Future use
Week	GPS week number	Unsigned short	2	
TOW	Predicted GPS time in seconds	Double	8	
UNICS_TT_SNR_PRN[12]	*See following*	Unsigned long	4	
• Bits 0-7 (8 bits) Pseudorandom noise; PRN is 0 if no data • Bits 8-15 (8 bits) Signal-to noise ratio (SNR); SNR=10.0 *log10* (0.8192*SNR) • Bits 16-23 (8 bits) PhaseTrackTime (PTT); in units of 1/10 sec; range=0 to 25 sec (if greater than 25 see UIDoppler_FL[12] below) • Bits 24-31 (8 bits) CycleSlip Counter (CSC); increments by 1 every cycle with natural rollover after 255				
UIDoppler_FL[12]	*See following*	Unsigned long	4	
• Bit 0 (1 bit) Phase; Location 0; 1 if valid (0 otherwise) • Bit 1 (1 bit) TrackTime; 1 if track time > 25.5 seconds (0 otherwise) • Bits 2-3 (2 bits) Unused • Bits 4-31 (28 bits) Doppler; Signed (two's compliment) Doppler in units of m/sec x 4096. (i.e., LSB=1/4096), range = +/- 32768 m/sec. Computed as phase change over 1/10 sec.				
PseudoRange[12]	Pseudorange	Double	8	

Phase[12]	Phase (m) L1 wave = 0.190293672798365	Double	8	

Structure

```
typedef struct
{
  SUnionMsgHeader  m_sHead;
  unsigned short   m_wSpare1;        /* spare 1 (zero)*/
  unsigned short   m_wWeek;          /* GPS Week Number */
  double           m_dTow;           /* Predicted GPS Time in seconds */
  SObservations    m_asObvs[CHANNELS_12];/* 12 sets of observations*/
  unsigned short   m_wCheckSum;      /* sum of all bytes of the datalength */
  unsigned short   m_wCRLF;          /* Carriage Return Line Feed */
} SBinaryMsg96;                      /* length = 8 + (300) + 2 + 2 = 312 */
```

Additional Information

Message has a BlockID of 96 and is 300 bytes, excluding the header and epilogue

Related Commands

JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin97 Message

Message Type Binary

Description Processor statistics

Command Format to Request Message

`$JBIN,97,r<CR><LF>`

where:

- '97' = Bin97 message
- 'r' = message rate in Hz (20, 10, 2, 1, 0, or .2)

Message Format

Message Component	Description	Type	Bytes	Values
CPUFactor	CPU utilization factor Multiply by 450e-06 to get percentage of spare CPU that is available **Note:** This field is only relevant on the old SLX platforms and Eclipse platform. It is not relevant for the Crescent receivers.	Unsigned long	4	Positive
MissedSubFrame	Total number of missed sub frames in the navigation message since power on	Unsigned short	2	Positive
MaxSubFramePnd	Max sub frames queued for processing at any one time	Unsigned short	2	Positive
MissedAccum	Total number of missed code accumulation measurements in the channel tracking loop	Unsigned short	2	Positive
MissedMeas	Total number missed pseudorange measurements	Unsigned short	2	Positive
Spare 1	Not used at this time	Unsigned long	4	Future use
Spare 2	Not used at this time	Unsigned long	4	Future use
Spare 3	Not used at this time	Unsigned long	4	Future use
Spare 4	Not used at this time	Unsigned short	2	Future use
Spare 5	Not used at this time	Unsigned short	2	Future use

Structure

```
typedef struct

{
```

```
  SUnionMsgHeader    m_sHead;
  unsigned long    m_ulCPUFactor;        /* CPU utilization Factor (%=multby 450e-6)
*/
  unsigned short m_wMissedSubFrame;      /* missed subframes */
  unsigned short m_wMaxSubFramePend; /* max subframe pending*/
  unsigned short m_wMissedAccum;         /* missed accumulations */
  unsigned short m_wMissedMeas;          /* missed measurements */
  unsigned long    m_ulSpare1;           /* spare 1 (zero)*/ unsigned
  long             m_ulSpare2;           /* spare 2 (zero)*/ unsigned
  long             m_ulSpare3;           /* spare 3 (zero)*/ unsigned
  short m_wSpare4; /* spare 4 (zero)*/ unsigned short m_wSpare5;    /*
  spare 5 (zero)*/
  unsigned short m_wCheckSum;            /* sum of all bytes of the datalength */
  unsigned short m_wCRLF;                /* Carriage Return Line Feed */
} SBinaryMsg97;                          /* length = 8 + (28) + 2 + 2 = 40 */
```

Additional Information

Message has a BlockID of 97 and is 28 bytes, excluding the header and epilogue

Related Commands

JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin98 Message

Message Type

Binary

Description

GPS satellite and almanac information

Command Format to Request Message

```
$JBIN,98,r<CR><LF>
```

where:

- '98' = Bin98 message
- 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
AlmanData[8]	SV data, 8 at a time	SSVAlmanData		See following
LastAlman	Last almanac processed	Byte	1	0 to 31
IonoUTCVFlag	Flag that is set when ionosphere modeling data is extracted from the GPS sub frame 4	Byte	1	0 = not logged 2 = valid
Spare	Not used at this time	Unsigned short	2	Future use

Structure

```
typedef struct
{
  SUnionMsgHeader  m_sHead;
  SSVAlmanData     m_asAlmanData[8];  /* SV data, 8 at a time */
  unsigned char    m_byLastAlman;     /* last almanac processed */
  unsigned char    m_byIonoUTCVFlag;  /* iono UTC flag */
  unsigned short   m_wSpare;          /* spare */
  unsigned short   m_wCheckSum;       /* sum of all bytes of the datalength */
  unsigned short   m_wCRLF;           /* Carriage Return Line Feed */
} SBinaryMsg98;                       /* length = 8 + (64+1+1+2) + 2 + 2 = 80 */
```

Additional Information

Message has a BlockID of 98 and is 68 bytes, excluding the header and epilogue

Related Commands

JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin99 Message

Message Type [Binary]

Description GPS L1 diagnostic information

Command Format to Request Message

```
$JBIN,99,r<CR><LF>
```

where:

- '99' = Bin99 message
- 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
NavMode	Navigation mode data (lower 3 bits hold the GPS mode, upper bit set if differential is available)	Byte	1	Lower 3 bits take on the values: 0 = time not valid 1 = No fix 2 = 2D fix 3 = 3D fix Upper bit (bit 7) is 1 if differential is available
UTCTimeDiff	Whole seconds between UTC and GPS time (GPS minus UTC)	Byte	1	Positive
GPSWeek	GPS week associated with this message	Unsigned short	2	0 to 65536
GPSTimeofWeek	GPS tow (sec) associated with this message	Double	8	0.0 to 604800.0
sChannelData[CHANNELS_12]	Channel data	SChannelData	12 x 24 = 288	
ClockErrAtL1	Clock error of the GPS clock oscillator at L1 frequency in Hz	Short	2	-32768 to 32768
Spare	Not used at this time	Unsigned short	2	Future use

Structure

```
typedef struct
{
  SUnionMsgHeader    m_sHead;
```

```
unsigned char    m_byNavMode;          /* Nav Mode FIX_NO, FIX_2D, FIX_3D
                                          (high bit =has_diff) */
char             m_cUTCTimeDiff;       /* whole Seconds between UTC and GPS        */
unsigned short m_wGPSWeek;             /* GPS week */
double           m_dGPSTimeOfWeek;     /* GPS tow    */
SChannelData     m_asChannelData[CHANNELS_12]; /* channel data */ short
                 m_nClockErrAtL1;      /* clock error at L1, Hz */
unsigned short m_wSpare;               /* spare */
unsigned short m_wCheckSum;            /* sum of all bytes of the datalength */
unsigned short m_wCRLF;                /* Carriage Return Line Feed */
} SBinaryMsg99;                           /* length = 8 + 304 + 2 + 2 = 316 */
```

Additional Information	Message has a BlockID of 99 and is 304 bytes, excluding the header and epilogue
Related Commands	JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin100 Message

Message Type	Binary
Description	GPS L2 diagnostic information

Command Format to Request Message

`$JBIN,100,r<CR><LF>`

where:

- '100' = Bin100 message
- 'r' = message rate in Hz (1 or 0)

Message Format

Message Component	Description	Type	Bytes	Values
NavMode	Navigation mode data (lower 3 bits hold the GPS mode, upper bit set if differential is available)	Byte	1	Lower 3 bits take on the values: 0 = time not valid 1 = No fix 2 = 2D fix 3 = 3D fix Upper bit (bit 7) is 1 if differential is available
UTCTimeDiff	Whole seconds between UTC and GPS time (GPS minus UTC)	Byte	1	Positive
GPSWeek	GPS week associated with this message	Unsigned short	2	0 to 65536
MaskSatsUsedL2P	L2P satellites used, bit mapped 0..31	Unsigned long		
GPSTimeofWeek	GPS tow (sec) associated with this message	Double	8	0.0 to 604800.0
MaskSatsUsedL1P	L1P satellites used, bit mapped 0..31	Unsigned long		
sChannelData[CHANNELS_12]	L2 channel data	SChannelData	12 x 24 = 288	

Structure

```
typedef struct
{
  SUnionMsgHeader m_sHead;
  unsigned char   m_byNavMode;          /* Nav Mode FIX_NO, FIX_2D, FIX_3D
                                          (high bit =has_diff) */
  char            m_cUTCTimeDiff;       /* whole Seconds between UTC and GPS        */
  unsigned short  m_wGPSWeek;           /* GPS week */
  unsigned long   m_ulMaskSatsUsedL2P   /* L2P SATS Used, bit mapped 0..31 */
  double          m_dGPSTimeOfWeek;     /* GPS tow    */
  unsigned long   m_ulMaskSatsUsedL1P; /* L1P SATS Used, bit mapped 0..31 */
  SChannelL2Data  m_asChannelData[CHANNELS_12];      /* channel data */
  unsigned short  m_wCheckSum;          /* sum of all bytes of the datalength */
  unsigned short  m_wCRLF;              /* Carriage Return Line Feed */
} SBinaryMsg100;                        /* length = 8 + 260 + 2 + 2 = 272 */
```

Additional Information

Message has a BlockID of 100 and is 260 bytes, excluding the header and epilogue

Related Commands

JBIN

Topic Last Updated: v1.06 / March 10, 2015

Bin209 Message

Message Type [Binary]

Description SNR and status for all GNSS tracks

Command Format to Request Message

```
$JBIN,209,r<CR><LF>
```

where:

- '209' = Bin209 message
- 'r' = message rate in Hz

Message Format

Message Component	Description	Type	Bytes	Values
GPSTimeofWeek	GPS tow (sec) associated with this message	Double	8	0.0 to 604800.0
GPSWeek	GPS week associated with this message	Unsigned short	2	0 to 65536
UTCTimeDiff	Whole Seconds between UTC and GPS	char	1	
Page	Bits 0-1 = Antenna: 0 = Master, 1 = Slave, 2 = Slave2 Bits 2-4 = Page ID: 0 = page 1, 1 = page 2, etc Bits 5-7 = Max page ID: 0 = only 1 page, 1 = 2 pages	Unsigned char	1	
sSVData	SNR data			

Structure

```
typedef struct
{
    SUnionMsgHeader    m_sHead;                //
    double             m_dGPSTimeOfWeek;       // GPS tow
    unsigned short     m_wGPSWeek;             // GPS week
    char               m_cUTCTimeDiff;         // Whole Seconds between UTC and GPS
    unsigned char      m_byPage;
                                      // Bits 0-1 = Antenna: 0 = Master, 1 = Slave, 2 = Slave2
                                      // Bits 2-4 = Page ID: 0 = page 1, 1 = page 2, etc
                                      // Bits 5-7 = Max page ID: 0 = only 1 page, 1 = 2 pages
    SSVSNRData         m_asSVData[40];         // SNR data
    unsigned short     m_wCheckSum;            // sum of all bytes of the header and data
    unsigned short     m_wCRLF;                // Carriage Return Line Feed
} SBinaryMsg209;                               // length = 8 + 332 + 2 + 2 = 344
```

Additional Information

Related Commands

JBIN

Topic Last Updated: v1.07 / Octoter 13, 2016

CRMSK Message

Message Type Beacon Receiver

Description Operational status message of SBX

Command Format to Request Message

```
$GPCRQ,MSK<CR><LF>
```

Message Format

```
$CRMSK,FFF.F,X,DDD,Y,N*CC<CR><LF>
```

where:

Message Component	Description
FFF.F	Frequency, in kHz (283.5 to 325)
X	Tune mode (M = manual, A = automatic)
DDD	MSK bit rate, in bps (100 or 200)
Y	MSK rate selection mode (M = manual, A = automatic)
N	Period of output of performance status message, in seconds (0 to 100); see CRMSS
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands GPCRQ,MSK

Topic Last Updated: v1.00 / August 11, 2010

CRMSS Message

Message Type Beacon Receiver

Description Performance status message of SBX

Command Format to Request Message

```
$GPCRQ,MSS<CR><LF>
```

Message Format

```
$CRMSS,XX,YY,FFF.F,DDD*CC<CR><LF>
```

where:

Message Component	Description
XX	Signal strength, in dB µV/m
YY	Signal-to-noise ratio, in dB
FFF.F	Frequency, in kHz (283.5 to 325)
DDD	MSK bit rate in bps (100 or 200)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands GPCRQ,MSS

Topic Last Updated: v1.00 / August 11, 2010

GLMLA Message

Message Type [GLONASS](#)

Description GLONASS almanac data

Contains complete almanac data for one GLONASS satellite. Multiple sentences may be transmitted, one for each satellite in the GLONASS constellation.

Command Format to Request Message

```
$JASC,GLMLA,r[,OTHER]<CR><LF>
```

where:

- **'r' = 1 (on) or 0 (off)**
  When set to on the message is sent once (one message for each tracked satellite at 1 second intervals) and then sent again whenever satellite information changes

- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format `$GLMLA,A.A,B.B,CC,D.D,EE,FFFF,GG,HHHH,IIII,JJJJJJ,KKKKKK,MMMMMM,NNNNNN,PPP,QQQ*hh<CR><LF>`

where:

Message Component	Description
A.A	Total number of sentences
B.B	Sentence number
CC	Satellite ID (satellite slot) number
D.D	Calendar day count within the four year period beginning with the previous leap year
EE	Generalized health of the satellite and carrier frequency number respectively
FFFF	Eccentricity
GG	DOT, rate of change of the draconitic circling time
HHHH	Argument of perigee
IIII	16 MSB of system time scale correction
JJJJJJ	Correction to the average value of the draconitic circling time
KKKKKK	Time of the ascension node, almanac reference time
MMMMMM	Greenwich longitude of the ascension node
NNNNNN	Correction to the average value of the inclination angle
PPP	LSB of system time scale correction
QQQ	Course value of the time scale shift

Example

Additional Information Similar to the GPS message GPALM

Related Commands JASC,GL

Topic Last Updated: v1.05 / January 18, 2013

GNSSPositionData Message

Message Type

NMEA 2000 CAN

Description

Detailed GPS position information

The GNSSPositionData message (PGN 0x1F805/129029) has an update rate of 1 Hz and DLC of 43, 47, or 51, dependent on the NumberOfReferenceStations.

Command Format to Request Message

Message is continuously output on the CAN port for the following products:

- A100, continuously output
- A325, continuously output when NMEA 2000 mode is enabled
- V102, continuously output when NMEA 2000 mode is enabled, requires NMEA 2000 adapter

Message Format

The following table provides the start bit, length (bit), value type, factor, and offset for the GNSSPositionData message.

Field Name	Start bit	Length (Bit)	Byte Order	Value Type	Factor
SequenceID	0	8	Intel	Unsigned	1
PositionDate	8	16	Intel	Unsigned	1
PositionTime	24	32	Intel	Unsigned	0.0001
LatitudeLow	56	32	Intel	Unsigned	1.00E-16
LatitudeHigh	88	32	Intel	Signed	4.29E-07
LongitudeLow	120	32	Intel	Unsigned	1.00E-16
LongitudeHigh	152	32	Intel	Signed	4.29E-07
AltitudeLow	184	32	Intel	Unsigned	1.00E-6
AltitudeHigh	216	32	Intel	Signed	4294.97
TypeOfSystem	248	4	Intel	Unsigned	1
GNSSMethod	252	4	Intel	Unsigned	1
GNSSIntegrity	256	2	Intel	Unsigned	1
GNSS_Reserved1	258	6	Intel	Unsigned	1
NumberOfSVs	264	8	Intel	Unsigned	1
HDOP	272	16	Intel	Signed	0.01
PDOP	288	16	Intel	Signed	0.01
GeodalSeparation	304	32	Intel	Signed	0.01
NumberOfReferenceStations	336	8	Intel	Unsigned	1
ReferenceStationType1	344	4	Intel	Unsigned	1

ReferenceStationID1	348	12	Intel	Unsigned	1
AgeOfDGNSSCorrections1	360	16	Intel	Unsigned	0.01
ReferenceStationType2	376	4	Intel	Unsigned	1
ReferenceStationID2	380	12	Intel	Unsigned	1
AgeOfDGNSSCorrections2	392	16	Intel	Unsigned	0.01

The following table provides the offset, minimum and maximum values, unit, and comment for the GNSSPositionData message.

Field Name	Offset	Min	Max	Unit	Comment
SequenceID	0	0	255		An upward counting number used to tie related information together between different PGNS
PositionDate	0	0	65532	day	Days since January 1, 1970. Date is relative to UTC time.
PositionTime	0	0	86401	sec	24 hour clock, 0=midnight, time is in UTC
LatitudeLow	0	0	4.29E-07	deg	Latitude referenced to WGS-84
LatitudeHigh	0	-90	90	deg	Latitude referenced to WGS-84
LongitudeLow	0	0		deg	Longitude referenced to WGS-84
LongitudeHigh	0	-180		deg	Longitude referenced to WGS-84
AltitudeLow	0	0		m	Altitude referenced to WGS-84
AltitudeHigh	0	-9.22 E+12		m	Altitude referenced to WGS-84
TypeOfSystem	0	0	4		0x0 GPS 0x1 GLONASS 0x2 GPS and GLONASS 0x3 GPS and SBAS, (WAAS/EGNOS) 0x4 GPS and SBAS and GLONASS

GNSSMethod	0	0	15		0x0 No GPS 0x1 GNSS fix 0x2 DGNSS fix 0x3 Precise GNSS 0x4 RTK fixed integer 0x5 RTK float 0x6 Estimated (DR) mode 0x7 Manual input 0x8 Simulate mode 0xE Error
GNSSIntegrity	0	0	3		0x0 No integrity checking 0x1 Safe 0x2 Caution 0X3 Unsafe
GNSS_Reserved1	0	0	63		
NumberOfSVs	0	0	252		Numeric count, event counter
HDOP	0	-327.64	327.64		Dilution of Precision (DOP) indicates the contribution of satellite configuration geometry to positioning error
PDOP	0	-327.64	327.64		Dilution of Precision (DOP) indicates the contribution of satellite configuration geometry to positioning error
GeodalSeparation	0	-2.15 E+07	2.15 E+07	m	The difference between the earth ellipsoid and mean sea-level (period), defined by the reference datum used in the position solution. '-' indicates mean sea-level below ellipsoid
NumberOfReferenceStations	0	0	252		Number of reference stations reported
ReferenceStationType1	0	0	15		0x0 GPS 0x1 GLONASS 0xE Error
ReferenceStationID1	0	0	4095		Reference station ID
AgeOfDGNSSCorrections1	0	0	655.32	sec	Age of differential corrections
ReferenceStationType2	0	0	15		0x0 GPS 0x1 GLONASS 0xE Error
ReferenceStationID2	0	0	4095		Reference station ID
AgeOfDGNSSCorrections2	0	0	655.32	sec	Age of differential corrections

Additional Information	
Related Commands	

Topic Last Updated: v1.00 / August 11, 2010

GNSSPositionRapidUpdates Message

Message Type

NMEA 2000 CAN

Description

Abbreviated GPS position information

The GNSSPositionRapidUpdates message (PGN 0x1F801/129025) has an update rate equal to the subscribed rate (default of 10 Hz) and DLC of 8.

Command Format to Request Message

Message is continuously output on the CAN port for the following products:

- A100, continuously output
- A325, continuously output when NMEA 2000 mode is enabled
- V102, continuously output when NMEA 2000 mode is enabled, requires NMEA 2000 adapter

Message Format

The following table provides the start bit, length (bit), value type, factor, and offset for fields of the GNSSPositionRapidUpdates message.

Field Name	Start bit	Length (Bit)	Byte Order	Value Type	Factor	Offset	Min	Max	Unit
Latitude	0	32	Intel	Signed	0.0000001	0	-90	90	deg
Longitude	32	32	Intel	Signed	0.0000001	0	-180	180	deg

Additional Information

Related Commands

Topic Last Updated: v1.00 / August 11, 2010

GPALM Message

Message Type Data

Description Message number (individual and total), week number, satellite health, and the almanac data for each satellite in the GPS constellation up to a maximum of 32 messages

Command Format to Request Message

```
$JASC,GPALM,r[,OTHER]<CR><LF>
```

where

- **'r' = 1 (on) or 0 (off)**
  When set to on the message is sent once (one message for each tracked satellite at 1 second intervals) and then sent again whenever satellite information changes
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without th brackets)

Message Format

```
$GPALM,A,B,C,D,E,F,G,H,J,K,L,M,N,P,Q*CC<CR><LF>
```

where:

Response Component	Description	As Displayed in First Full Line of Example Below This Table
A	Total number of messages	31
B	Message number	1
C	Satellite PRN number	02
D	GPS week number (0-1023)	1617
E	Satellite health (bits 17-24 of message)	00
F	Eccentricity	50F6
G	Reference time of almanac (TOA)	0F
H	Satellite inclination angle (sigma)	FD98
J	Rate of right ascension (omega dot)	FD39
K	Square root of semi-major axis (root A)	A10CF3
L	Perigee (omega)	81389B
M	Ascending node longitude (omega O)	423632
N	Mean anomaly (mo)	BD913C
P	Clock parameter 0 (af0)	148
Q	Clock parameter 1 (af1)	001
*CC	Checksum	
<CR>	Carriage return	

<LF>	Line feed	

Example

```
$>

$GPALM,31,1,02,1617,00,50F6,0F,FD98,FD39,A10CF3,81389B,423632,BD913C,148,001*

$GPALM,31,2,03,1617,00,71B9,0F,F6C2,FD45,A10C96,2B833C,131DB4,BA69EE,2B1,001*

$GPALM,31,3,04,1617,00,4F01,0F,FD03,FD39,A10BFC,1C6C35,42EDB1,35B537,112,003*

$GPALM,31,4,05,1617,00,121B,0F,08C8,FD61,A10C5C,09CA99,6D7257,021B32,79F,7FE*

$GPALM,31,5,06,1617,00,337F,0F,FB6B,FD49,A10CC2,DBE103,161127,10CD11,18C,7FE*

.

.

.

$GPALM,31,29,30,1617,00,6A85,0F,0ADD,FD5C,A11A83,3F6243,EBCC46,E8548D,145,001

$GPALM,31,30,31,1617,00,4037,0F,1778,FD3E,A10C28,D62817,C32ADF,781125,01B,001

$GPALM,31,31,32,1617,00,65B5,0F,0956,FD65,A10DD0,DD74BA,71125D,985AE3,751,7FE
```

Additional Information

Similar to the GLONASS message GLMLA

Related Commands

JASC,GP

Topic Last Updated: v1.05 / January 18, 2013

GPDTM Message

Message Type Data

Description Datum reference

Command Format to Request Message

```
$JASC,GPDTM,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate (in Hz) of (1 or 0)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPDTM,CCC,A,X.X,K,X.X,L,X.X,CCC*CC<CR><LF>
```

where:

Message Component	Description
CCC	Local datum (normally W84, but could be NAD83 when using beacon in North America)
A	Local datum subdivision code
X.X	Latitude offset, in minutes
K	Latitude indicator; value is N (North latitude) or S (South latitude)
X.X	Longitude offset, in minutes
L	Longitude indicator; value is E (East longitude) or W (West longitude)
X.X	Altitude offset, in meters
CCC	Reference datum (always W84)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example

```
$GPDTM,W84,,0.0,N,0.0,E,0.0,W84*CC<CR><LF>
```

Additional Information

Related JASC,GP

Commands

Topic Last Updated: v1.04 / May 29, 2012

GPGGA Message

Note: **This topic provides information pertaining to GPS. The format is the same for the messages pertaining to GNSS and GLONASS (see Additional Information below).**

Message Type Data

Description Detailed GNSS position information (most frequently used NMEA 0183 data message)

Command Format to Request Message

```
$JASC,GPGGA,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate (in Hz) of 20, 10, 5, 4, 2, 1, 0, or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPGGA,HHMMSS.SS,DDMM.MMMMM,K,DDDMM.MMMMM,L,N,QQ,PP.P,AAAA.AA,M,±XX.XX,M,
SSS,RRRR*CC<CR><LF>
```

where:

Message Component	Description
HHMMSS.SS	UTC time in hours, minutes, and seconds of the position
DDMM.MMMMM	Latitude in degrees, minutes, and decimal minutes (you can set the number of decimal places using the JNP command)
K	Latitude indicator; value is N (North latitude) or S (South latitude)
DDDMM.MMMMM	Longitude in degrees, minutes, and decimal minutes (you can set the number of decimal places using the JNP command)
L	Longitude indicator; value is E (East longitude) or W (West longitude)
N	Quality indicator; value is: • 0 = no position • 1 = undifferentially corrected position (autonomous) • 2 = differentially corrected position (SBAS, DGPS,Atlas DGPS service, L-Dif and e-Dif) • 4 = RTK fixed integer (Crescent RTK, Eclipse RTK),Atlas high precision services converged • 5 = RTK float,Atlas high precision services converging
QQ	Number of satellites used in position solution
P.P	Horizontal dilution of precision (HDOP)

A.A	Antenna altitude, in meters, re: mean-sea-level (geoid)
M	Units of antenna altitude (M = meters)
G.G	Geoidal separation (in meters)
M	Units of geoidal separation (M = meters)
SSS	Age of differential corrections, in seconds
RRRR	Differential reference station ID
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example `$GPGGA,001038.00,3334.2313457,N,11211.0576940,W,2,04,5.4,354.682,M,-26.574,M,7.0,0138*79`

Additional Information

This message provides information specific to the satellite system identified by the first two characters of the message.

GPGGA - GPS information
GNGGA - GNSS information
GLGGA - GLONASS information

The JNMEA,GGAALLGNSS command significantly affects the output of the GGA message. If you are tracking more than GNSS signals, Hemisphere GNSS highly recommends that you review this command.

Related Commands

JASC,GP, JASC,GN, JASC,GL, JNMEA,GGAALLGNSS

Topic Last Updated: v1.07 / February 16, 2017

GPGLL Message

Note: **This topic provides information pertaining to GPS. The format is the same for the messages pertaining to GNSS and GLONASS (see Additional Information below).**

Message Type	Data
Description	Latitude and longitude data
Command Format to Request Message	`$JASC,GPGLL,r[,OTHER]<CR><LF>` where: • 'r' = message rate in Hz of 20, 10, 2, 1, 0, or .2 (0 turns off the message) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)
Message Format	`$GPGLL,DDMM.MMMMM,S,DDDMM.MMMMM,S,HHMMSS.SS,S*CC<CR><LF>` where:

Message Component	Description
DDMM.MMMMM	Latitude in degrees, minutes, and decimal minutes
S	S = N (North latitude) or S (South latitude)
DDDMM.MMMMM	Longitude in degrees, minutes, and decimal minutes
S	S = E (East longitude) or W (West longitude)
HHMMSS.SS	UTC time in hours, minutes, and seconds of GNSS position
S	Status, S = A (valid) or V (invalid)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information	This message provides information specific to the satellite system identified by the first two characters of the message. GPGLL - GPS information GNGLL - GNSS information GLGLL - GLONASS information The JNMEA,GGAALLGNSS command significantly affects the output of the GLL message. If you are tracking more than GNSS signals, Hemisphere GNSS highly recommends that you review this command.
Related Commands	JASC,GP, JASC,GN, JASC,GL, JNMEA,GGAALLGNSS

Topic Last Updated: v1.07 / February 16, 2017

GPGNS Message

Note: **This topic provides information pertaining to GPS. The format is the same for the messages pertaining to GNSS and GLONASS (see Additional Information below).**

Message Type Data

Description Fixes data for single or combined (GPS, GLONASS, possible future satellite systems, and systems combining these) satellite navigation systems

Command Format to Request Message

`$JASC,GPGNS,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate (in Hz) of 20, 10, 2, 1, 0, or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and the brackets) and enacts a change on the other port when you send the command with it (without the brack

Message Format

`$GPGNS,HHMMSS.SS,DDMM.MMMMM,K,DDDMM.MMMMM,L,MM,QQ,H.H,A.A,G.G,D.D,R.R,NS*CC<C`

where:

Message Component	Description
HHMMSS.SS	UTC time in hours, minutes, and seconds of the position
DDMM.MMMMM	Latitude in degrees, minutes, and decimal minutes (you can set the number of decimal places using the JNP command)
K	Latitude indicator; value is N (North latitude) or S (South latitude)
DDDMM.MMMMM	Longitude in degrees, minutes, and decimal minutes (you can set the number of decimal places using the JNP command)
L	Longitude indicator; value is E (East longitude) or W (West longitude)
MM	Mode indicator Variable length valid character field type with the first two characters currently defined. • First character indicates the use of GPS satellites • Second character indicates the use of GLONASS satellites If another satellite system is added to the standard, the mode indicator will be extended to three characters. New satellite systems shall always be added on the right, so the order of characters in the Mode Indicator is: GPS, GLONASS, other satellite systems in the future. The characters shall take one of the following values: • N = No fix. Satellite system not used in position fix, or fix not valid • A = Autonomous. Satellite system used in non-differential mode in position fix • D = Differential. Satellite system used in differential mode in position fix • P = Precise. Satellite system used in precision mode. Precision mode is defined as no deliberate degradation (such as Selective Availability) and higher resolution code (P-code) is used to compute position fix. • R = Real Time Kinematic. Satellite system used in RTK mode with fixed integers • F = Float RTK. Satellite system used in real time kinematic mode with floating integers

	• E = Estimated (dead reckoning) mode • M = Manual input mode • S = Simulator mode The mode indicator shall not be a null field.
QQ	Number of satellites used in position solution
P.P	Horizontal dilution of precision (HDOP)
A.A	Antenna altitude, in meters, re: mean-sea-level (geoid)
G.G	Geoidal separation (in meters)
SSS	Age of differential corrections, in seconds
RRRR	Differential reference station ID
NS	Navigational status; options are: • S = Safe • C = Caution • U = Unsafe • V = Not valid for navigation
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example `$GPGNS,224749.00,3333.4268304,N,11153.3538273,W,D,19,0.6,406.110,-26.294,6.0,0138,S,*6A`

Additional Information

This message provides information specific to the satellite system identified by the first two characters of the message.

GPGNS - GPS information
GNGNS - GNSS information
GLGNS - GLONASS information
GAGNS – GALILEO information

The JNMEA,GGAALLGNSS command significantly affects the output of the GNS message. If you are tracking more than GNSS sign Hemisphere GNSS highly recommends that you review thiscommand.

Related Commands

JASC,GP, JASC,GN, JASC,GL, JNMEA,GGAALLGNSS

Topic Last Updated: v1.07/ February 16, 2017

GPGRS Message

Message Type Data

Description Supports Receiver Autonomous Integrity Monitoring (RAIM)

Command Format to Request Message

```
$JASC,GPGRS,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 1, 0, or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPGRS,HHMMSS.SS,M,X.X ... X.X,GSID,SID*CC<CR><LF>
```

where:

Message Component	Description
HHMMSS.SS	UTC time
M	Mode: 0 = residuals used to calculate the position given in the GPGGA or GPGNS message 1 = residuals were recomputed after the GPGGA or GPGNS message position was computed
X.X ... X.X	Range residuals, in meters, for satellites used in the navigation solution. Order must match order of satellite ID numbers in GPGSA message. When GPGRS message is used, the GPGSA and GPGSV messages are generally required with this message.
GSID	GNSS system ID, value is 1 (GPS)
SID	Signal ID, value is 1 (L1 C/A)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands JASC,GP

Topic Last Updated: v1.04 / May 29, 2012

GNGSA Message

Note: **This topic provides information pertaining to all GNSS constellations. The format is the same for the messages pertaining to only GPS and GLONASS (see Additional Information below).**

Message Type

Data

Description

DOP and active satellite information

Only satellites used in the position computation are present in this message. Null fields are present when data is unavailable due to the number of satellites tracked.

Command Format to Request Message

```
$JASC,GNGSA,r[,OTHER]<CR><LF>
```

```
where:
```

'r' = message rate in Hz of 1 or 0 (0 turns off the message)

',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GNGSA,A,B,CC ... OO,P.P,Q.Q,R.R,GSID*CC<CR><LF>
```

where:

Message Component	Description
A	Satellite acquisition mode (M = manually forced to 2D or 3D, A = automatic swap between 2D and 3D)
B	Position mode (1 = fix not available, 2 = 2D fix, 3 = 3D fix)
CC to OO	Satellites used in the position solution, a null field occurs if a channel is unused
P.P	Position Dilution of Precision (PDOP) = 1.0 to 9.9
Q.Q	Horizontal Dilution of Precision (HDOP) 1.0 to 9.9
R.R	Vertical Dilution of Precision (VDOP) = 1.0 to 9.9
GSID	GNSS system ID, value is 1 (GPS), 2 (GLONASS), 3 (GALILEO), 5 (BEIDOU)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

This message provides information specific to the satellite system(s) identified by the first two characters of the message.
GNGSA - GNSS information (all constellations)
GPGSA - GPS information
GLGSA - GLONASS information

Related Commands

JASC,GP, JASC,GN, JASC,GL

Topic Last Updated: v1.07 / February 16, 2017

GPGST Message

Message Type Data

Description GNSS pseudorange error statistics and position accuracy

Command Format to Request Message

`$JASC,GPGST,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate in Hz of 1 or 0 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

`$GPGST,HHMMSS.SS,A.A,B.B,C.C,D.D,E.E,F.F,G.G*CC<CR><LF>`

where:

Message Component	Description
HHMMSS.SS	UTC time in hours, minutes, and seconds of the GPS position
A.A	Root mean square (rms) value of the standard deviation of the range inputs to the navigation process. Range inputs include pseudoranges and differential GNSS (DGNSS) corrections.
B.B	Standard deviation of semi-major axis of error ellipse, in meters
C.C	Standard deviation of semi-minor axis of error ellipse, in meters
D.D	Error in Eclipse’s semi major axis origination, in decimal degrees, true north
E.E	Standard deviation of latitude error, in meters
F.F	Standard deviation of longitude error, in meters
G.G	Standard deviation of altitude error, in meters
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands

JASC,GP

Topic Last Updated: v1.01 / September 23, 2010

GPGSV Message

Note: **This topic provides information pertaining to GPS. The format is the same for the message pertaining to other constelations (see Additional Information below).**

Message Type Data

Description GNSS satellite in view

Null fields occur where data is unavailable due to the number of satellites tracked.

Command Format to Request Message

```
$JASC,GPGSV,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 1 or 0 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPGSV,T,M,N,II,EE,AAA,SS,…II,EE,AAA,SS,SID*CC<CR><LF>
```

where:

Message Component	Description
T	Total number of messages
M	Message number (1 to 3)
N	Total number of satellites in view
II	Satellite number
EE	Elevation, in degrees (0 to 90)
AAA	Azimuth (true), in degrees (0 to 359)
SS	Signal strength, in dB-Hz (0 - 99) To compare with SNR values found in Bin messages (such as Bin96) subtract 30 from this signal strength value for an approximate SNR value SS - 30 = SNR (from Bin message)
SID	Signal ID, value is 1 (L1 C/A)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

This message provides information specific to the satellite system identified by the first two characters of the message.

GPGSV – GPS information
GLGSV – GLONASS information

GAGSV – GALILEO information

GQGSV – QZSS information

If you request GNGSV the receiver will respond with GPGSV messages only.

Related Commands	JASC,GP, JASC,GL, BEIDOU

Topic Last Updated: v1.07 / February 16, 2017

GPHDG/HEHDG Message

Message Type Data

Description

Magnetic deviation and variation for calculating magnetic or true heading

The message simulates data from a magnetic sensor although it does not actually contain one. The purpose of this message is to support older systems that may not be able to accept the HDT message that is recommended for use.

Command Format to Request Message

```
$JASC,GPHDG,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 20, 10, 2, 1, 0 or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPHDG,s.s,d.d,D,v.v,V*CC<CR><LF>
```

or

```
$HEHDG,s.s,d.d,D,v.v,V*CC<CR><LF>
```

where:

Message Component	Description
s.s	Magnetic sensor reading, in degrees
d.d	Magnetic deviation, in degrees
D	E = Easterly deviation, W = Westerly deviation
v.v	Magnetic variation, in degrees
V	E = Easterly deviation, W = Westerly deviation
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

You can change the HDG message header to either GP or HE using the JATT,NMEAHE command.

Related Commands

JASC,GP

Topic Last Updated: v1.00 / August 11, 2010

GPHDM/HEHDM Message

Message Type

Data

Description

Magnetic heading of the vessel derived from the true heading calculated

Command Format to Request Message

`$JASC,GPHDM,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate in Hz of 20, 10, 2, 1, 0 or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

`$GPHDM,X.X,M*CC<CR><LF>`

or

`$HEHDM,X.X,M*CC<CR><LF>`

where:

Message Component	Description
X.X	Current heading, in degrees
M	Indicates magnetic heading
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

You can change the HDM message header to either GP or HE using the JATT,NMEAHE command.

Related Commands

JASC,GP

Topic Last Updated: v1.02 / January 25, 2011

GPHDT/HEHDT Message

Message Type

Data

Description

True heading of the vessel

This is the direction that the vessel (antennas) is pointing and is not necessarily the direction of vessel motion (the course over ground).

Command Format to Request Message

```
$JASC,GPHDT,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 20, 10, 2, 1, 0 or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPHDT,X.X,T*CC<CR><LF>
```

or

```
$HEHDT,X.X,T*CC<CR><LF>
```

where:

Message Component	Description
X.X	Current heading, in degrees
T	Indicates true heading
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

You can change the HDT message header to either GP or HE using the JATT,NMEAHE command.

Related Commands

JASC,GP

Topic Last Updated: v1.00 / August 11, 2010

GPHEV Message

Message Type	Data
Description	Heave value in meters
Command Format to Request Message	`$JASC,GPHEV,1<CR><LF>`
Message Format	`$GPHEV,H,*CC<CR><LF>` where:
Additional Information	
Related Commands	JASC,GP

Message Component	Description
H	Heave value, in meters
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Topic Last Updated: v1.00 / August 11, 2010

GPRMC Message

Message Type Data

Description Contains recommended minimum specific GNSS data

Command Format to Request Message

```
$JASC,GPRMC,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 10, 2, 1, 0, or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPRMC,HHMMSS.SS,A,DDMM.MMM,N,DDDMM.MMM,W,Z.Z,Y.Y,DDMMYY,D.D,V,M,NS*CC<CR><LF
```

where:

Message Component	Description
HHMMSS.SS	UTC time in hours, minutes, and seconds of the GPS position
A	Status (A = valid, V = invalid)
DDMM.MMM	Latitude in degrees, minutes, and decimal minutes
N	Latitude location (N = North latitude, S = South latitude)
DDDMM.MMM	Longitude in degrees, minutes, and decimal minutes
W	Longitude location (E = East longitude, W = West longitude)
Z.Z	Ground speed, in knots
Y.Y	Track made good, reference to true north
DDMMYY	UTC date of position fix in day, month, and year
D.D	Magnetic Variation, in degrees
V	Variation sense (E = East, W = West)
M	Mode indicator Variable length valid character field type with the first two characters currently defined. • First character indicates the use of GPS satellites If another satellite system is added to the standard, the mode indicator will be extended to three characters. New satellite systems shall always be added on the right, so the order of characters in the Mode Indicator is: GPS, GLONASS, other satellite systems in the future. The characters shall take one of the following values: • N = No fix. Satellite system not used in position fix, or fix not valid • A = Autonomous. Satellite system used in non-differential mode in position fix

	• D = Differential. Satellite system used in differential mode in position fix • P = Precise. Satellite system used in precision mode. Precision mode is defined as no deliberate degradation (such as Selective Availability) and higher resolution code (P-code) is used to compute position fix. • R = Real Time Kinematic. Satellite system used in RTK mode with fixed integers • F = Float RTK. Satellite system used in real time kinematic mode with floating integers • E = Estimated (dead reckoning) mode • M = Manual input mode • S = Simulator mode The mode indicator shall not be a null field.
NS	Navigational status; options are: • S = Safe • C = Caution • U = Unsafe • V = Not valid for navigation
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands JASC,GP

Topic Last Updated: v1.04 / May 29, 2012

GPROT/HEROT Message

Message Type

Data

Description

Vessel’s rate of turn (ROT) information

Command Format to Request Message

```
$JASC,GPROT,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 20, 10, 2, 1, 0 or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPROT,X.X,A*CC<CR><LF>
```

or

```
$HEROT,X.X,A*CC<CR><LF>
```

where:

Message Component	Description
X.X	Rate of turn in °/min (negative when the vessel bow turns to port)
A	Flag indicating the data is valid
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

You can change the ROT message header to either GP or HE using the JATT,NMEAHE command.

Related Commands

JASC,GP

Topic Last Updated: v1.00 / August 11, 2010

GPRRE Message

Message Type

Data

Description

Satellite range residuals and estimated position error

Command Format to Request Message

```
$JASC,GPRRE,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 1 or 0 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPRRE,N,II,RR ... II,RR,HHH.H,VVV.V*CC<CR><LF>
```

where:

Message Component	Description
N	Number of satellites used in position computation
II	Satellite number
RR	Range residual, in meters
HHH.H	Horizontal position error estimate, in meters
VVV.V	Vertical position error estimate, in meters
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands

JASC,GP

Topic Last Updated: v1.00 / August 11, 2010

GPVTG Message

Message Type Data

Description Course over ground and ground speed

Command Format to Request Message

`$JASC,GPVTG,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate in Hz of 20, 10, 2, 1, 0, or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

`$GPVTG,TTT,T,MMM,M,NNN.NN,N,KKK.KK,K,X*CC<CR><LF>`

where:

Message Component	Description
TTT	True course over ground (COG) in degrees (000 to 359)
T	True course over ground indicator (always 'T')
MMM	Magnetic course over ground in degrees (000 to 359)
M	Magnetic course over ground indicator (always 'M')
NNN.NN	Speed over ground in knots
N	Speed over ground in knots indicator (always 'N')
KKK.KK	Speed over ground in km/h
K	Speed over ground in km/h indicator (always 'K')
X	Mode A = Autonomous mode D = Differential mode E = Estimated (dead reckoning) mode M = Manual input mode S = Simulator mode N = Data not valid
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example Sample message output:

`$GPVTG,103.85,T,92.79,M,0.14,N,0.25,K,D*1E`

Additional Information	
Related Commands	JASC,GP

Topic Last Updated: v1.00 / August 11, 2010

GPZDA Message

Message Type	Data
Description	UTC time and date information
Command Format to Request Message	`$JASC,GPZDA,r[,OTHER]<CR><LF>` where: • 'r' = message rate in Hz of 20, 10, 2, 1, 0, or .2 (0 turns off the message) • ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$GPZDA,HHMMSS.SS,DD,MM,YYYY,XX,YY*CC<CR><LF>
```

where:

Message Component	Description
HHMMSS.SS	UTC time in hours, minutes, and seconds of the GPS unit
DD	Day (0 to 31)
MM	Month (1 to 12)
YYYY	Year
XX	Local zone description in hours (-13 to 13)
YY	Local zone description in minutes (0 to 59)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands JASC,GP

Topic Last Updated: v1.00 / August 11, 2010

NMEACogSogData Message

Message Type NMEA 2000 CAN

Description

GPS speed and direction information

The NMEACogSogData command (PGN 0x1F802/129026) has an update rate equal to the subscribed rate (default of 10 Hz) and DLC of 8.

Command Format to Request Message

Message is continuously output on the CAN port for the following products:

- A100, continuously output
- A325, continuously output when NMEA 2000 mode is enabled
- V102, continuously output when NMEA 2000 mode is enabled, requires NMEA 2000 adapter

Message Format

The following table describes the fields of the NMEACogSogData message:

Field Name	Start Bit	Length (Bit)	Byte Order	Value Type	Factor	Min	Max	Comment
NMEA_SequenceID	0	8	Intel	Unsigned	1	0	255	An upward counting number used to tie related information together between different PGNs
NMEA_Direction Reference	8	2	Intel	Unsigned	1	0	3	0x0 True north 0x1 Magnetic north 0x2 Error 0X3 Null
NMEA_Reserved1	10	6	Intel	Unsigned	1	0	63	
NMEA_Course OverGround	16	16	Intel	Unsigned	0.0001	0	6.5535	GPS based travel direction, in rad
NMEA_Speed OverGround	32	16	Intel	Unsigned	0.01	0	655.35	GPS based travel speed, in m/s

Additional Information

Related Commands

Topic Last Updated: v1.00 / August 11, 2010

PASHR Message

Message Type Vector, Data

Description Time, true heading, roll, pitch, and heave data in one message

Command Format to Request Message

```
$JASC,PASHR,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate (in Hz) of 20, 10, 5, 4, 2, 1, 0, or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without th brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Message Format

```
$PASHR,hhmmss.ss,HHH.HH,T,RRR.RR,PPP.PP,heave,rr.rrr,pp.ppp,hh.hhh,QF*CC<CR><
```

where:

Message Component	Description
hhmmss.ss	UTC time
HHH.HH	Heading value in decimal degrees
T	True heading (T displayed if heading is relative to true north)
RRR.RR	Roll in decimal degrees (- sign will be displayed when applicable)
PPP.PP	Pitch in decimal degrees (- sign will be displayed when applicable)
heave	Heave, in meters
rr.rrr	Roll standard deviation in decimal degrees
pp.ppp	Pitch standard deviation in decimal degrees
hh.hhh	Heading standard deviation in decimal degrees
QF	Quality Flag • 0 = No position • 1 = All non-RTK fixed integer positions • 2 = RTK fixed integer position
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional

Information	
Related Commands	JASC,PASHR

Topic Last Updated: v1.05 / January 18, 2013

PSAT,ATTSTAT Message

Message Type Data,

Description

Command Format to Request Message

`$JASC,PSAT,ATTSTAT,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate in Hz of 1 or 0 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

`$PSAT,ATTSTAT,S,MSEP,CSEP,Heading,TYPE,Pitch,Roll,Q,N,SYS,NUM,SNR,*CC`

where:

Message Component	Description
S	ID of the secondary antenna
MSEP	custom separation between antennas Manually entered (when the value is MOV, it means MOVEBASE is on)
CSEP	auto GPS antenna separation
Heading	Heading
TYPE	Heading indicator, value is: N= Heading used GNSS G=Heading used gyroscope
Pitch	pitch
Roll	roll
Q	The current setting of antenna directivity, value is P= antennas placed front and back, output pitch R= antennas placed left and right, output roll
N	The number of satellite used by the secondary antenna
SYS	Systems in use: • GPS: L1, L2, L5 • GLONASS: G1, G2 • BDS: B1,B2 B3 • Galileo: E5a, E5b, E5a+b, E6
NUM	Number of satellites used by each system

SNR	Quality of each SNR path, where: • A is > 20 dB • B is > 18 dB • C is > 15 db • D is <= 15 dB

*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example

$PSAT,ATTSTAT,1,MOV,0.504,334.75,N,1.71,8.0,P,30,(,L1,L2,G1,G2,B1,B2,B3,)(,12,10,9,9,10,10,0,)(, A,A,C,B,B,B,D,)*1D

Additional Information Issuing the JSAVE command after setting JASC,PSAT,ATTSTAT to 1 (message on at 1Hz) does not save this setting. You must JASC,PSAT,ATTSTAT (set it to 1) each time you power on the receiver.

Related Commands and Messages JASC,PSAT,ATTSTAT command

Topic Last Updated: v1.07.1 / April 27, 2017

PSAT,GBS Message

Message Type Data

Description Used to support Receiver Autonomous Integrity Monitoring (RAIM)

Command Format to Request Message

```
$JASC,GPGBS,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 1 or 0 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$PSAT,GBS,HHMMSS.SS,KK.K,LL.L,AA.A,ID,P.PPPPP,B.B,S.S,FLAG,GSID,SID*CC<CR><LF
```

where:

Message Component	Description
HHMMSS.SS	UTC time in hours, minutes, and seconds of the GGA or GNS fix associated with this sentence
KK.K	Expected error in latitude
LL.L	Expected error in longitude
AA.A	Expected error in altitude
ID	ID number of most likely failed satellite
P.PPPPP	Probability of HPR fault
B.B	Estimate of range bias, in meters, on most likely failed satellite
S.S	Standard deviation of range bias estimate
FLAG	Based on horizontal radius: 0 = Good 1 = Warning 2 = Bad or Fault
GSID	GNSS system ID, value is 1 (GPS)
SID	Signal ID, value is 1 (L1 C/A)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands	JASC,GP

Topic Last Updated: v1.04 / May 29, 2012

PSAT,HPR Message

Message Type

Data

Description

Proprietary NMEA message that provides the true heading, pitch, roll, and time in a single message

During normal operation heading and pitch are derived from GPS and roll comes from the inertial sensor. While coasting heading is based on gyro and pitch/roll are from the inertial sensor.

Command Format to Request Message

`$JASC,GPHPR,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate in Hz of 20, 10, 2, 1, 0 or .2 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

`$PSAT,HPR,TIME,HEADING,PITCH,ROLL,TYPE*CC<CR><LF>`

where:

Message Component	Description
TIME	UTC time (HHMMSS.SS)
HEADING	Heading (degrees)
PITCH	Pitch (degrees)
ROLL	Roll (degrees)
TYPE	N = GPS derived heading G = gyro heading
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands

JASC,GP

Topic Last Updated: v1.05 / January 18, 2013

PSAT,INTLT Message

Message Type

[Data](#)

Description

Proprietary NMEA message that provides the tilt measurements from the internal inclinometers in degrees. It delivers an output of crude accelerometer measurements of pitch and roll with no temperature compensation or calibration for GPS heading/pitch/roll.

Pitch and roll are factory calibrated over temperature to be accurate to ±3°C.

CAUTION: User calibration will clear out precise factory calibration.

Command Format to Request Message

```
$JASC,INTLT,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 1 or 0 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$PSAT,INTLT,PITCH,ROLL*CC<CR><LF>
```

where:

Message Component	Description
PITCH	Pitch (degrees)
ROLL	Roll (degrees)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Additional Information

Related Commands

[JASC,GP](#)

Topic Last Updated: v1.00 / August 11, 2010

PSAT,BLV Message

Message Type Data, Local Differential and RTK

Description Contains RTK fixprogress information

Command Format to Request Message

```
$JASC,PSAT,BLV,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 1 or 0 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$PSAT,HHMMSS.SS,DATE,A.A,B.B,C.C,ID,STATE,number,pdop*CC<CR><L F>
```

where

Message Component	Description
HHMMSS.SS	UTC time (HHMMSS.SS)
DATE	Date (day-month-year)
A.A	North component of base to rover vector (m)
B.B	Esat component of base to rover vector (m)
C.C	Up component of base to rover vector (m)
ID	Base station ID
STATE	Quality indicator; value is: • 0 = no position • 1 = undifferentially corrected position (autonomous) • 2 = differentially corrected position (SBAS, DGPS, Atlas DGPS service, L-Dif and e-Dif) • 4 = RTK fixed integer (Crescent RTK, Eclipse RTK) ,Atlas high precision services converged 5 = RTK float, Atlas high precision services converging
NUMBER	Number of used satellite
PDOP	PDOP
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example	$PSAT,BLV,000151.00,051115,-0.001,0.002,-0.003,0333,4,20,1.2*52
Additional Information	
Related Commands	JASC, PSAT, BLV

Topic Last Updated: v1.07 / February 16, 2017

PSAT,FVI Message

Message Type Data, Local Differential and RTK

Description Contains much more special information

Command Format to Request Message

```
$JASC,PSAT,FVI,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate in Hz of 0,1,2,5,10,20 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

```
$PSAT,FVI,HHMMSS.SS, DDMM.MMMM, DDDMM.MMMM, AA.AAA,
E.E,F.F,G.G,HHH.HHH,hh.hhh,PP.PP,pp.ppp,RR.RRR,rr.rrr,ve.eee,v
n.nnn,vu.uuu,vv.vvv,LE.EEE,LN.NNN,LU.UUU,ZONE,UEEE.EEEE,UNNN.N
NNN,PN,SN,p,h,L,sss*CC<CR><LF>
```

where

Message Component	Description
HHMMSS.SS	UTC time
DDMM.MMMM	Latitude in degrees, minutes, and decimal minutes
DDMM.MMMM	Longitude in degrees, minutes, and decimal minutes
AA.AAA	altitude
E.E	Standard deviation of latitude error, in meters
F.F	Standard deviation of longitude error, in meters
G.G	Standard deviation of altitude error, in meters
HHH.HHH	Heading (degrees)
hh.hhh.	Standard deviation of heading error, in degrees
PP.PP	Pitch (degrees)
pp.ppp	Standard deviation of pitch error, in degrees
RR.RRR	Roll (degrees)
rr.rrr	Standard deviation of roll error, in degrees
Ve.eee	East to speed（m/s）
Vn.nnn	North to speed (m/s)
Vu.uuu	Vertical speed (m/s)
Vv.vvv	Speed over ground (m/s)

LE.EEE	East component of master to slave vector (m)
LN.NNN	North component of master to slave vector (m)
LU.UUU	Up component of master to slave vector (m)
ZONE	projection area
UEEE.EEEE	East to positon of projection area
UNNN.NNNN	North to position of projection area
PN	Number of satellites used by the primary antenna
SN	Number of satellites used by the secondary antenna
P	Position indicator; value is: • 0 = no position • 1 = undifferentially corrected position (autonomous) • 2 = differentially corrected position (SBAS, DGPS ,Atlas DGPS service, L-Dif and e-Dif) • 4 = RTK fixed integer (Crescent RTK, Eclipse RTK), Atlas high precision services converged • 5 = RTK float, Atlas high precision services converging
H	Heading indicator; value is: • 0 = no heading or heading is invalid • 1 = heading is valid
L	Distance between base and rover in meter
SSS	Age of differential corrections, in seconds
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example $PSAT,FVI,011657.00,40.071345258,116.326680384,51.2922,0.001,0.003,0.003,28.358,0.106,-5.306,0.087,,,0.030,-0.001,-0.062,0.030,-0.001,0.001,-0.002,117.0,442562.296,4437668.138,25,26,4,1,4.759,1*6B

Additional Information

Related Commands JASC,PSAT,FVI

Topic Last Updated: v1.07 / February 16, 2017

PSAT,RTKPROG Message

Message Type Data, Local Differential and RTK

Description Contains RTK fixprogress information

Command Format to Request Message

`$JASC,PSAT,RTKPROG,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate in Hz of 1 or 0 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

`$PSAT,RTKPROG,,R,F,N,SS1,SS2,SS3,MASK*CC<CR><LF>`

where

Message Component	Description
R	1 = Ready to enter RTK ambiguity fix 0 = Not ready to enter RTK ambiguity fix
F	1 = Receiver running in RTK ambiguity fix mode 0 = Receiver not running in RTK ambiguity fix mode
N	Number of satellites used to fix
SS1	summer-1 SS1 must be significantly larger than SS2 and SS3 to enter R=1 mode
SS2	summer-2
SS3	summer-3
MASK	Bit mask; bits identify which GNSS observables are being received from base recently (1 = GPS, 3 = GPS + GLONASS)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example `$PSAT,RTKPROG,1,1,24,243.0,0.0,0.0,3*4F<CR><LF>`

- Ready to enter RTK ambiguity fix
- Receiver running in RTK ambiguity fix mode
- 24 satellites used to fix

- summer-1 is 243.0, summer-2 is 0, summer-3 is 0
- Bit mask is 3 (GPS + GLONASS)

Additional Information	Issuing the JSAVE command after setting JASC,PSAT,RTKPROG to 1 (message on at 1Hz) does not save this setting. You must enable JASC,PSAT,RTKPROG (set it to 1) each time you power on the receiver.
Related Commands	JASC,PSAT,RTKPROG

Topic Last Updated: v1.04 / May 29, 2012

PSAT,RTKSTAT Message

Message Type	Data, Local Differential and RTK
Description	Contains the most relevant parameters affecting RTK

Command Format to Request Message

`$JASC,PSAT,RTKSTAT,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate in Hz of 1 or 0 (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

`$PSAT,RTKSTAT,MODE,TYP,AGE,SUBOPT,DIST,SYS,NUM,SNR,RSF,BSF,HAG,ACCSTAT,SNT*CC`

where:

Message Component	Description
MODE	Mode (FIX,FLT,DIF,AUT,NO)
TYP	Correction type (DFX,ROX,CMR,RTCM3,CMR+,...)
AGE	Age of differential corrections, in seconds
SUBOPT	Subscription code (see Interpreting the $JI and $JK 'Date'/Subscription Codes to determine the meaning of the subscription code)
DIST	Distance to base in kilometers
SYS	Systems in use: • GPS: L1, L2, L5 • GLONASS: G1, G2 • BDS: B1,B2 B3 • Galileo: E5a, E5b, E5a+b, E6
NUM	Number of satellites used by each system
SNR	Quality of each SNR path, where: • A is > 20 dB • B is > 18 dB • C is > 15 db • D is <= 15 dB
RSF	Rover slip flag (non zero if parity errors in last 5 minutes, good for detecting jamming and TCXO issues)

BSF	Base slip flag
HAG	Horizontal accuracy guess
ACCSTAT	RTK accuracy status (hex), where: • 0x1 = no differential or differential too old, for the application • 0x2 = problems with differential message • 0x4 = horizontal position estimate poor for the application • 0x8 = HDOP high, poor satellite geometry • 0x10 = fewer than 6 L1 sats used • 0x20 = poor L1 SNRs • 0x40 = not in RTK mode • 0x80 = not in RTK mode or RTK only recently solved (< 10 secs ago) • 0x100 = RTK solution compromised, may fail The status message can be any of the above or any combination of the above. For example, a status message of '047' indicates the following: • 0x1 = no differential or differential too old, for the application • 0x2 = problems with differential message • 0x4 = horizontal position estimate poor for the application • 0x40 = not in RTK mode
SNT	Ionospheric scintillation, values are: • 0 (little or no scintillation - does not adversely affect RTK solution) • 1-100 (scintillation detected - adversely affects RTK solution)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example

`$PSAT,RTKSTAT,FIX,ROX,1,007F,9.5,(,L1,L2,G1,G2,)(,14,11,9,9,)(,A,A,A,A,),0,1,0.011,000`

- Fixed mode
- ROX corrections
- Diff age = 1 second
- Subscribed options = 7F (see Understanding Additive Codes for information on subscriptions)
- Distance to base = 9.5 km
- L1,L2,G1,G2 are the systems in use

- Satellites used: L1 = 14, L2 = 11, G1 = 9, G2 = 9
- SNR quality is (> 20 dB), (> 20 dB), (> 20 dB), (> 20dB)
- Rover slip flag = 0
- Base slip flag = 1
- Horizontal accuracy guess = 0.011
- RTK accuracy status = 000 (no issues or errors)
- Little or no ionospheric scintillation

Additional Information	Issuing the JSAVE command after setting JASC,PSAT,RTKSTAT to 1 (message on at 1Hz) does not save this setting. You must e JASC,PSAT,RTKSTAT (set it to 1) each time you power on the receiver.
Related Commands and Messages	JASC,PSAT,RTKSTAT command JQUERY,RTKSTAT message

Topic Last Updated: v1.07 / Octoter 13, 2016

PSAT,VCT Message

Message Type Data, Local Differential and RTK

Description

Command Format to Request Message

`$JASC,PSAT,VCT,r[,OTHER]<CR><LF>`

where:

- 'r' =0,1,2,5,10,20HZ (0 turns off the message)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets)

Message Format

`$PSAT,VCT,ID,HHMMSS.SS,A.A,B.B,C.C,D,E.E,F.F,G.G,H.H*CC<CR><LF>`

where

Message Component	Description
ID	antenna pair ID (always 1 for now)
HHMMSS.SS	UTC time in hours, minutes, and seconds of the position
A.A	Heading in degree
B.B	Pitch in degree
C.C	Roll in degree
N	Normal, not coasting
E.E	distance between antennas (m)
F.F	North component of master to slave vector (m)
G.G	East component of master to slave vector (m)
H.H	Up component of master to slave vector (m)
*CC	Checksum
<CR>	Carriage return
<LF>	Line feed

Example `$PSAT,VCT,1,011657.00,28.358,-5.306,,N,4.7591,4.1530,2.2823,- 0.4401*1F`

Additional Information	
Related Commands	JASC,PSAT,VCT

Topic Last Updated: v1.07 / Octoter 13, 2016

RD1 Message

Message Type Data

Description SBAS diagnostic information

Command Format to Request Message

`$JASC,D1,r[,OTHER]<CR><LF>`

where:

- 'r' = message rate (0 = Off, 1 = On at 1Hz)
- ',OTHER' = optional field, enacts a change in the RD1 message on the current port when you send the command without it (and without the brackets) and enacts a change in the RD1 message on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Message Format

`$RD1,SEC,WEEK,FREQ,DSPLOCK,BER2,AGC,DDS,DOPPLER,DSPSTAT,ARMSTAT,DIFFSTAT,NAVCON<CR><LF>`

where:

Message Component	Description
SEC	Second of GPS week (may be a couple of seconds old)
WEEK	GPS week number
FREQ	L-band frequency in MHz (1575.4200 is used for SBAS)
DSPLOCK	N/A
BER2	BER - given for both SBAS satellites being tracked
AGC	L-band signal strength
DDS	0.0 for SBAS
DOPPLER	0 for SBAS

<table>
<tr><td>DSPSTAT</td><td>Status bit mask for the DSP tracking of SBAS

Bit 0 = Carrier lock
Bit 1 = BER OK (Viterbi lock) (yellow LED 2)
Bit 2 =Atlas: DSP got lock and has stable freq; WAAS: Frame sync2
Bit 3 = Frame sync1
Bit 4 = Track mode (same as carrier lock)
Bits 5 - 15 Unused
</td></tr>
<tr><td>ARMSTAT</td><td>Status bit mask for the ARM GPS solution (ARM status values shown below)

Bit 0 = GPS lock (yellow LED 1)
Bit 1 = DGPS valid data
Bit 2 = ARM has lock
Bit 3 = Diff and GPS (flashing green LED 3)
Bit 4 = GPS solution is good (solid green LED 3)
Bit 5 = ARM controls yellow LED 2
Bit 6 = ARM command for yellow LED 2
Bits 7 - 15 Unused
</td></tr>
<tr><td>DIFFSTAT</td><td>SBAS PRN of the satellite in use</td></tr>
<tr><td>NAVCON</td><td>Series of hex character fields with each field representing the number of GPS satellites satisfying a certain condition, all of which conditions are required if the satellite is to be used in the solution
Example of NAVCON for the value 179889A shown below (read right to left)
<table>
<tr><th>Hex Field</th><th>Description</th><th>Value</th></tr>
<tr><td>1 (right most field)</td><td>Hexadecimal count of satellites with valid tracks</td><td>A</td></tr>
<tr><td>2</td><td>Hexadecimal count of satellites for which an ephemeris message has been received</td><td>9</td></tr>
<tr><td>3</td><td>Hexadecimal count of satellites which are healthy</td><td>8</td></tr>
<tr><td>4</td><td>Hexadecimal count of satellites which passed the criteria of hex fields 1,2,3 and 5 (satellites that er tracked, have an ephemeris, are healthy, and are above the elevation mask)</td><td>8</td></tr>
<tr><td>5</td><td>Hexadecimal count of satellites above the elevation mask</td><td>9</td></tr>
<tr><td>6</td><td>Hexadecimal count of satellites for which a differential correction is available</td><td>7</td></tr>
<tr><td>7</td><td>Hexadecimal count of satellites for which a differential correction is NOT available</td><td>1</td></tr>
</table></td></tr>
<tr><td><CR></td><td>Carriage return</td></tr>
</table>

<LF>	Line feed

Additional Information	
Related Commands	JASC,D1 (RD1)

Topic Last Updated: v1.07 / February 16, 2017

TSS1 Message

Message Type Vector, Data

Description Heave, pitch, and roll message in the commonly used TSS1 message format

Command Format to Request Message

```
$JASC,PTSS1,r[,OTHER]<CR><LF>
```

where:

- 'r' = message rate (in Hz) of 0 (off), 0.25, 0.5, 1, 2, 4, 5, 10, or 20 (if subscribed)
- ',OTHER' = optional field, enacts a change on the current port when you send the command without it (and without the brackets) and enacts a change on the other port when you send the command with it (without the brackets). See Configuring the Data Message Output for detailed information on 'THIS' and 'OTHER' port terminology.

Message Format

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

where:

Message Component	Description
XX	Horizontal acceleration (hex value), in 3.83 cm/s², with a range of zero to 9.81 m/s²
AAAA	Vertical acceleration (hex value - 2's complement), in 0.0625 cm/s², with a range of –20.48 to +20.48 m/s²
S	Space character
M	Space if positive; minus if negative
HHHH	Heave, in centimeters, with a range of –99.99 to +99.99 meters
Q	Status flag Value — Description h — Heading aided mode (settling) - The System is receiving heading aiding signals from a gyrocompass but is still awaiting the end of the three minutes settling period after power-on or a change of mode or heave bandwidth. The gyrocompass takes approximately five minutes to settle after it has been powered on. During this time, gyrocompass aiding of the System will not be perfect. The status flag does NOT indicate this condition. F — Full aided mode (settled condition) - The System is receiving and using aiding signals from a gyrocompass and from a GNSS receiver or a Doppler log.
M	Space if positive; minus if negative
RRRR	Roll, in units of 0.01 degrees (ex: 1000 = 10°), with a range of –99.99° to +99.99°
S	Space character
M	Space if positive; minus if negative

PPPP	Pitch, in units of 0.01 degrees (ex: 1000 = 10°), with a range of –99.99° to +99.99°
<CR>	Carriage return
<LF>	Line feed

Example

:020010 -0001F 0023 -0169

where:

- XX = 02, horizontal acceleration, which is 7.66 cm/s²
  (XX = 02 (hex) = decimal 2, multiplied by 3.83 cm/s² yields 7.66 cm/s²)
- AAAA = 0010, vertical acceleration, which is 1 cm/s²
  (AAAA = 0010 (hex), which = decimal 16, multiplied by 0.0625 cm/s² yields 1 cm/s²)
- S = (space)
- M = (minus), meaning following heave value is negative
- HHHH = 0001, heave, which is 1 cm (-1 cm based on the M value)
- Q = F, status flag, which is full aided mode
- M = (space), meaning following roll value is positive
- RRRR = 0023, roll, which is 0.23°
- S = (space)
- M = (minus), meaning following pitch value is negative
- PPPP = 0169, pitch, which is 1.69°

Additional Information

Related Commands [JASC,PTSS1]

Topic Last Updated: v1.07 / February 16, 2017

Resources

Reference Documents

National Marine Electronics Association

National Marine Electronics Association (NMEA) Standard for Interfacing Marine Electronic Devices

Version 2.1, October 15, NMEA 1995

7 Riggs Avenue

Severna Park, MD 21146 Tel:

+1-410-975-9425

Tel Toll Free: +1-800-808-6632

http://www.nmea.org/

Radio Technical Commission for Maritime Services

RTCM Recommended Standards for Differential NAVSTAR GPS Service Version 2.2

Developed by Special Committee No. 104, RTCM 1998 1800

N Kent St, Suite 1060

Arlington, VA 22209, USA Tel:

+1-703-527-2000

http://www.rtcm.org/

Radio Technical Commission for Aeronautics

Minimum Operational Performance Standards (MOPS) for Global Positioning System/Wide Area Augmentation System Airborne Equipment

Document RTCA D0-229A, Special Committee No. 159, RTCA 1998 71828 L

Street, NW, Suite 805

Washington, D.C. 20036 USA Tel:

+1-202-833-9339

http://www.rtca.org/

ARIC Research Corporation

Interface Control Document, Navstar GPS Space Segment/Navigation User Interfaces

ICD-GPS-200, April 12, 2000

2250 E. Imperial Highway, Suite 450 El

Segundo, CA 90245-3509

http://www.navcen.uscg.gov/

Topic Last Updated: v1.02 / January 25, 2011

Websites

Hemisphere GNSS

http://www.hemispheregnss.com

FAA WAAS

This site offers general information on the WAAS service provided by the U.S. FAAS.

http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/waas/

ESA EGNOS System Test Bed

This site contains information relating to past performance, real-time performance, and broadcast schedule of EGNOS.

http://www.esa.int/esaNA/egnos.html

Solar and Ionosphereic Activity

The following sites are useful in providing details regarding solar and ionospheric activity.

http://iono.jpl.nasa.gov

http://www.spaceweather.com

Topic Last Updated: v1.06 / March 10, 2015

Change History v1.07.1

$PSAT, ATTSTAT	Added “S,” to $PSAT, ATTSTAT,MSEP, CSEP

Change History v1.07

Atlas Commands	Removed: •"L-Band Commands" Updated: •"Command/Description" table
Bin16	Moved from GALILEO Commands to GNSS Commands
Ethernet Configuration	New topic added
GALILEO Commands	Added note: *Note: For observations in tracking status, see GNSS, Bin 16 & Bin 19. Added: •Commands and Messages
GPS to GNSS	**Changed GPS to GNSS throughout the document where applicable**
GPGSA	Changed to GNGSA (where applicable)
GNGSA	Document system ID 6
GQGSV	Section added
JASC Command Overview	Removed: •JASC, PSAT,SMARTBASE Added: •JASC< PSAT< VCT,1 JASC, GG Command section added Replaced JASC, GP to JASC GN
JBIN Command	Added Bin16, Bin44
JBOTT Command	Removed: •"Omni", Added "L-Band". Removed from description "It also allows you to reset the L-band high precision services resolution algorithm."
JDIFFX, GNSSOUT	

Command	Added: •BEIDOU, GALILEO to Command Format. Replaced “both” GPOS and GLONASS with “all”GPS and GLONASS
JDIFFX, INCLUDE Command	Added: •[, ATLAS] to Command Format
JDIFF Subscription Code	Removed
JETHERNET-	Added topic
JETHERNET MODE	Added topic
JETHERNET PORTI	Added topic
JHP	Removed topic
JHP, LIMIT Command	Removed topic
JHP, MODE, AUTOSEED Command	Removed topic
JHP, MODE< IGNORECONV Command	Removed topic
JHP, POS \Command	Removed topic
JHP, POS, LAT, LON, HGT Command	Removed topic
JHP, POS, LAT, LON, HGT,,,, OTHER Command	Removed topic
JHP, POS, OTHER Command	Removed topic

JHP, POS, PRESENT Command	Removed topic
JHP, RESET, ACCURACY Command	Removed topic
JHP, RESET, ENGINE Command	Removed topic
JHP, SEED Command	Removed topic
JHP, SEED, LAT, LON, HGT Command	Removed topic
JHP, STATIC Command	Removed topic
JHP, STATUS, AUTOSEED Command	Removed topic
JLX BEAM Command	Added:•Receiver Response commands
JOMS Command	Removed topic
JPRN, EXCLUDE Command	Added: • : 'z,z,z…' represents the GALILEO PRNs you want to exclude", "Exclude no GALILEO PRNs: $JPRN,EXCLUDE,GAL,NONE<CR><LF>"
JSIGNAL Command	Description rewritten to read: "Set the receiver to use the specify signal: GNSS signals that the receiver will attempt to track. Specific signals shown here are only valid for receivers supporting the signal in question." Added: •$>JSIGNAL,INCLUDE[,L1CA][,L1P][,L2P][,L2C][,G1][,G2][,E1BC][,B1][,B2][,B3][,E5B][,QZSL1CA][,QZSL2C]<CR><LF> Changed Command Format to read: •Specify the signal(s) to be used $JSIGNAL,INCLUDE[,L1CA][,L1P][,L2P][,L2C][,G1][,G2][,E1BC][,B1][,B2][,B3][,E5B][,QZSL1CA][,QZSL2C][,ALL]<CR><LF> Specify the signal(s) NOT to be used

	$JSIGNAL,EXCLUDE[,L1CA][,L1P][,L2P][,L2C][,G1][,G2][,E1BC][,B1][,B2][,B3][,E5B][,QZSL1CA][,QZSL2C][,ALL]<CR><LF>
L-Band to Atlas	Changed L-Band to Atlas throughout the document (excluding commands)
L-Band	Removed: •high-precision, and high precision with GLONASS services” •4.L-band (DGPS) Replaced: “DGPS” with “Atlas”
NMEA 0183 Message	Format Updated to: XX NMEA 0183 talker field (GP = GPS, GL = GLONASS, GA = GALILEO, GB = BEIDOU, GN = All constellations)
Post-Processing	Added: the following messages, which must be logged in a binary file: Observations: Bin 76 (GPS), Bin 66 (GLONASS), Bin 36 (BEIDOU) Or Bin 16 (All constellations; required for GALILEO) Ephemeris: Bin 95 (GPS), Bin 65 (GLONASS), Bin 35 (BEIDOU), Bin 45 (GALILEO) Time conversion: Bin 94 (GPS), Bin 34 (BEIDOU), Bin 44 (GALILEO) Changed: (Crescent receivers must log Bin 94, 95, and 96 messages for GPS). Depending on the application, the binary data can be logged to a file and then translated to RINEX at a later time on a PC.
QZSS Commands and Messages	Section added
Using RIGHTARM to Load Firmware	Re-numbered list for accuracy

Change History v1.06

Binary Message	added the Bin3 and Bin209 message to the table.
Bin3	new topic
Bin209	new topic
Data Messages	Added the PSAT,ATTSTAT message
JASC Overview	Added the following command to the table: JASC,PSAT,ATTSTAT JASC,PSAT,BLV JASC,PSAT,FVI JASC,PSAT,SMARTBASE JASC,PSAT,VCT:
JASC,PSAT,ATTSTAT	new topic
JASC,PSAT,BLV	new topic
JASC,PSAT,FVI	new topic
JASC,PSAT,SMARTBASE	new topic
JASC.PSAT,VCT	new topic
JATT	Added the JATT,MOVEBASE command

JATT,MOVEBASE	new topic
JK,SHOW	new topic
JEPHOUT	new topic
JMODE Overview	Added the following command to the table: JMODE,BDSOFF JMODE,GLOOFF JMODE,GPSOFF JMODE,SURVEY JMODE,STRICKTRTK
JMODE,BDSOFF	new topic
JMODE,GLOOFF	new topic
JMODE,GPSOFF	new topic
JPPS	new topic, including JPPS,WIDTH command and JPPS,FREQ command
JPPS,WIDTH	new topic
JPPS,FREQ	new topic
JQUERY,TEMPERATURE	new topic
JRAD Overview	add the JRAD,10command
JRAD,10	new topic

JRTCM,INCLUDE	Updated Command format section
JRTCM,EXCLUDE	Updated Command format section
JSIGNAL	new topic
PSAT,RTKSTAT	Updated description for 'SYS' value in Message Format t to the following: SYS Systems in use: • GPS: L1, L2, L5 • GLONASS: G1, G2 • BDS: B1,B2 B3 • Galileo: E5a, E5b, E5a+b, E6

Change History v1.04

Beacon Receiver Commands and Messages	Merged topic with 'NMEA 0183 SBX Queries' topic
Bin1	Updated 'VEast' description to say "m/s" and not 'n/s"
Bin94	Updated description for 'r' value in Command Format to Request Message section to the following: 'r' = 1 (on) or 0 (off) When set to on the message is sent once and then sent again whenever satellite information changes
Bin95	Updated description for 'r' value in Command Format to Request Message section to the following: 'r' = 1 (on) or 0 (off) When set to on the message is sent once (one message for each tracked satellite at 1 second intervals) and then sent again whenever satellite information changes
Data Messages	Added the PSAT,RTKPROG message
General Operation and Configuration Commands	Added the following commands: • JDIFF,AVAILABLE • JFORCEAPP • JMODE,BASE • JMODE,FIXLOC • JMODE,GLOFIX • JMODE,SBASNORTK • JMODE,SURETRACK • JPRN,EXCLUDE • JSHOW,ASC • JSHOW,BIN • JSHOW,CONF • JSHOW,GP
GLMLA	Removed 'JASC' from beginning of response (just after '$') in Message Format section

GPALM	Updated Message format section
GPCRQ,MSK	Changed Command Type to Beacon Receiver
GPCRQ,MSS	Changed Command Type to Beacon Receiver
GPDTM	Updated Message format table for consistency
GPGGA	Updated Message format table for consistency
GPGNS	Updated Message format table for consistency and added "NS" field (navigational status)
GPGRS	Added "GSID" field (GNSS system ID) and "SID" field (signal ID) to Message Format section; also changed max output rate to 1 (so in the Command Format to Request Message section, instead of "20, 10, 2, 1, 0 or .2" it now says "1, 0 or .2")
GPGSA	Added "GSID" field (GNSS system ID) to Message Format section
GPGSV	Added "SID" field (signal ID) to Message Format section
GPRMC	Added "M" field (mode indicator) and "NS" field (navigational status) to Message Format section
JASC,GN	Corrected "MSG" column entries to begin with "GN" instead of "GP"
JASC,GP	Changed max output rate for GPGRS to 1 (so in the Command Format to Request Message section, instead of "20, 10, 2, 1, 0 or .2" it now says "1, 0 or .2")
JASC,PSAT,RTKPROG	New topic
JASC,PSAT,RTKSTAT	In Additional Information section removed incorrect text stating "To query the receiver for the current setting, issue the JSHOW command."
JATT,COGTAU	Added following paragraph in Description section: "COG is computed using only the primary GPS antenna (when using a multi-antenna system) and its accuracy depends upon the speed of the vessel (noise is proportional to 1/speed). This value is invalid when the vessel is stationary, as tiny movements due to calculation inaccuracies are not representative of a vessel's movement."
JATT,HRTAU	Changed heading rate time constant to rate of turn (ROT) time constant
JATT,HTAU	Updated Description section
JATT,PBIAS	In Additional Information section added text after first sentence

JATT,PTAU	Updated Description section
JDIFF	Updated Receiver Response section to show SOURCE and TYPE
JDIFF,AVAILABLE	New topic
JDIFFX,EXCLUDE	Added information for querying current setting
JDIFFX,GNSSOUT	Throughout topic replaced "GNSS output in correction formats" with "GNSS systems to be output in the differential"
JDIFFX,INCLUDE	Added information for querying current setting
JDIFFX,TYPE	Updated 'type' options in Receiver Response section
JFORCEAPP	New topic
JFREQ	UpdatedAtlas satellite table
JMODE Overview	Added the following commands: • JMODE,BASE • JMODE,FIXLOC • JMODE,GLOFIX • JMODE,SBASNORTK • JMODE,SURETRACK
JMODE	Added receiver responses for BASE, FIXLOC, GLOFIX, SBASNORTK, and SURETRACK
JMODE,BASE	New topic
JMODE,FIXLOC	New topic
JMODE,GLOFIX	New topic
JMODE,MIXED	Corrected query responses: • $>JMODE,MIXED,**ON** changed to $>JMODE,MIXED,**YES** • $>JMODE,MIXED,**OFF** changed to $>JMODE,MIXED,**NO**

JMODE,SBASNORTK	New topic
JMODE,SBASR	Corrected query responses: • $>JMODE,SBASR,**ON** changed to $>JMODE,SBASR,**YES** • $>JMODE,SBASR,**OFF** changed to $>JMODE,SBASR,**NO**
JMODE,SURETRACK	New topic
JMODE,TIMEKEEP	Corrected query responses: • $>JMODE,TIMEKEEP,**ON** changed to $>JMODE,TIMEKEEP,**YES** • $>JMODE,TIMEKEEP,**OFF** changed to $>JMODE,TIMEKEEP,**NO**
JMODE,TUNNEL	Corrected query responses: • $>JMODE,TIMEKEEP,**ON** changed to $>JMODE,TIMEKEEP,**YES** • $>JMODE,TIMEKEEP,**OFF** changed to $>JMODE,TIMEKEEP,**NO**
JNMEA,PRECISION	Added GPGNS to list of messages (in Description section) for which you can set the decimal places output
JNP	Added GPGNS to list of messages (in Description section) for which you can set the decimal places output
JPRN,EXCLUDE	New topic
JQUERY,RTKPROG	New topic
JRAD,7	Updated Receiver Response from `$>` to `$>JRAD,7,OK`
JRAD,9	New name of previous JRAD,9,1,1 command. Added information on "JRAD,9,0" that turns base mode off
JRAD,9,1,1	Changed command name to JRAD,9
JSHOW,ASC	New topic
JSHOW,BIN	New topic
JSHOW,CONF	New topic (some of the information in this topic appeared in the previous JSHOW topic)

JSHOW,GP	New topic (some of the information in this topic appeared in the previous JSHOW topic)
JSMOOTH	Added 'DEFAULT' to Command Format section and moved response text (regarding SHORT and LONG) from Command Format section to Receiver Response section
Local Differential and RTK Commands	Added the following commands and message: • JASC,PSAT,RTKPROG • JQUERY,RTKPROG • PSAT,RTKPROG
NMEA 0183 SBX Queries	Merged topic with Beacon Receiver Commands and Messages
PCSI,0	Changed Command Type to link to Beacon Receiver topic
PCSI,1	Changed Command Type to link to Beacon Receiver topic
PCSI,1,1	Moved example from Receiver Response section to new Example section
PCSI,2	Added Example section and changed Command Type to link to Beacon Receiver topic
PCSI,3,1	Changed Command Type to link to Beacon Receiver topic
PCSI,3,2	Added `$PCSI,ACK,3,2` as first line of receiver response
PCSI,3,3	Added `$PCSI,ACK,3,3` as first line of receiver response
PCSI,4	New topic
PCSI,5	New topic
PCSI,6	New topic
PCSI,7	New topic
PSAT,GBS	Added "GSID" field (GNSS system ID) and "SID" field (signal ID) to Message Format section
PSAT,RTKPROG	New topic

PSAT,RTKSTAT	Updated "ACCSTAT" field (accuracy status), added "SNT" field (ionospheric scintillation) field, and removed CMR+ from TYP (will show as CMR) in Message Format section; also added text regarding JSAVE in Additional Information section

Topic Last Updated: v1.04 / May 29, 2012

Change History v1.03

Bin66	**Updated description to refer to GLONASS L1/L2 instead of just GLONASS L1**
Bin69	**Updated description to refer to GLONASS L1/L2 instead of just GLONASS L1**
Bin76	Fixed spelling error: • In Description section under "To determine L1P or L2P", changed "buts" to "bits" in step 1 to read "Use the lower 16 bits provided in the message." • In Message format section corrected spelling errors:(1) changed "port" to "part" for Carrier Phase (High part) in both instances of P7_Doppler_FL row, and (2) changed "Cide" to "Code" in both instances of CodeAndPhase row
Bin98	**Added "GPS" to description**
Binary Messages	Updated descriptions for the following in the message table: • Bin66 - changed GLONASS L1 to GLONASS L1/L2 • Bin69 - changed GLONASS L1 to GLONASS L1/L2 • Bin98 - added "GPS" to description
Eclipse II Subscription Codes	Fixed spelling errors: • Removed redundant column on far right of table • Changed "eDiff" to "e-Dif" • Changed "Raw Ou" to "Raw Out" • Updated table formatting
General Operation and Configuration Commands	Added JSHOW,THISPORT command
GPGNS	Updated description of "mm" field (mode indicator) in Message Format section
Hardware Platforms Overview	New topic
Interpreting the $JI and $JK	Changed shading at bottom of topic to only shade '3000', not '01/01/3000'

'Date'/Subscription Codes

JALT	Updated Command Format, Receiver Response, and Example sections to more clearly define 'h' value.
JASC,PTSS1	Removed checksum; added units for Heave, Pitch, and Roll; changed "gyrocompass settle time" in 'h' description from "several hours" to "approximately five minutes" (see similar change in TSS1 message)
JBIN	Updated descriptions for the following in the message table: • Bin66 - changed GLONASS L1 to GLONASS L1/L2 • Bin69 - changed GLONASS L1 to GLONASS L1/L2 • Bin98 - added "GPS" to description
JDIFFX,TYPE	Corrected Receiver Response from `$>JDIFF,type` to `$>JDIFFX,TYPE,type` and added 'type' list
JHP,MODE,AUTOSEED	Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure."
JHP,MODE,IGNORECONV	Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure."
JHP,POS,LAT,LON,HGT,,,OTHER	Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure."
JHP,POS,OTHER	Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure."
JHP,STATUS,AUTOSEED	Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure." Added definitions for 'status' field in Receiver Response section
JK	Added 'DowngradeCode' field to Receiver Response section and updated response descriptions and Example section accordingly
JMODE,NULLNMEA	Corrected responses (in Receiver Response and Example sections): • Changed `$>JMODE,NULLNMEA,ON` to `$>JMODE,NULLNMEA,YES` • Changed `$>JMODE,NULLNMEA,OFF`

```
to $>JMODE,NULLNMEA,NO
```

JRTK Overview	Added JRTK,18,BEARING and JRTK,18,NEU topics
JRTK,18,BEARING	New topic
JRTK,18,NEU	New topic
JSHOW	Updated JFREQ (line 15 in table) to add information on bit rate and AUTO
JSHOW,THISPORT	New topic
JT	Added Receiver Response information for miniEclipse
Local Differential and RTK Commands	Added JRTK,18,BEARING and JRTK,18,NEU commands
Quick Start	New topic
PCSI,3,2	Updated Receiver Response format and table to include time and date
PSAT,RTKSTAT	Added description for ACCSTAT (accuracy status) parameter in message response
RD1	Removed checksum (*CC) from message format
TSS1	Removed checksum; added units for Heave, Pitch, and Roll; changed "gyrocompass settle time" in 'h' description from "several hours" to "approximately five minutes" (see similar change in JASC,TSS1 command)
Understanding Additive Codes	Updated examples at bottom of topic
Universal Development Kit	New topic

Topic Last Updated: v1.03 / January 11, 2012

Change History v1.02

Beacon Messages	Removed topic; information now part of Beacon Receiver Commands and Messages
Beacon Receiver Commands and Messages	Added information from now-removed Beacon Messages topic
Bin62	New topic
Bin65	New topic
Bin66	New topic
Bin69	New topic
Bin89	New topic
Bin98	AlmanData[8] is now a single row in the Message Format table with a link to SSVAlmanData that provides detailed information
Bin99	sChannelData[CHANNELS_12] is now a single row in the Message Format table with a link to SChannelData that provides detailed information
Binary Messages	Added information on new messages (Bin62, Bin65, Bin66, Bin69, Bin89)
Binary Messages Code	Updated to current version of code
COAST Technology	Added **Note** regarding Crescent and Eclipse boards
Commands and Messages Overview	Added link for GLONASS Commands and Messages and removed link for Beacon Messages since that information is now part of Beacon Receiver Commands and Messages
Configuring the Data Message Output	Added **Note** regarding specifying Port T when writing to a USB flash drive
Crescent Vector Commands and Messages	Added PASHR and TSS1 to Crescent Vector messages table
Data Messages	Added the GPALM message
Determining the Receiver Type and Current Application	Updated both tables

DGPS Base Station Commands	Updated definitions of commands
e-Dif Commands	Updated definitions of commands
EGNOS	Changed reference to "three satellites" to now read "multiple geostationary satellites and a network of ground stations"
General Operation and Configuration Commands	Added the following messages to the table: JDIFFX,EXCLUDE JDIFFX,GNSSOUT JDIFFX,INCLUDE JDIFFX,SOURCE JDIFFX,TYPE JFLASH,DIR JFLASH,FILE,CLOSE JFLASH,FILE,NAME JFLASH,FILE,OPEN JFLASH,FREESPACE JFLASH,NOTIFY,CONNECT JFLASH,QUERYCONNECT JMODE JMODE,FOREST JMODE,GPSONLY JMODE,L1ONLY JMODE,MIXED JMODE,NULLNMEA JMODE,SBASR JMODE,TIMEKEEP JMODE,TUNNEL JRELAY
GLMLA	New topic
GLONASS Commands and Messages	New topic
GPALM	New topic
GPCRQ,MSK	Added Example section
GPCRQ,MSS	Added Example section
GPGGA	Added Note at top of topic, updated Command Format to Request Message section to add "5" and "4" to list of message rates, added Example section, and updated Additional Information and Related Commandssections
GPGLL	Added Note at top of topic and updated Additional Information and Related

	Commands sections
GPGNS	Added Note at top of topic and updated Additional Information and Related Commands sections
GPGSA	Added Note at top of topic and updated Additional Information and Related Commands sections
GPGSV	Added Note at top of topic and updated Additional Information and Related Commands sections
GPHDM	Corrected alternate format in Message Format section from `$HCHDM,x.x,M*CC<CR><LF>` to `$HEHDM,x.x,M*CC<CR><LF>`
GPMSK	Updated Receiver Response section and added Example section
JAGE	Added Example section and updated Additional Information section
JAIR	Added query code to Command Format section and added Example section
JALT	Added "SATS" as a "c" value in the Command Format section, added Example section, and updated Additional Information section
JAPP	Categorized Command Format section by receiver type, moved receiver response text from Command Format section to Receiver Response section, and added Example section
JASC Overview	Added the following messages to the table: JASC,CMR JASC,GL JASC,GN JASC,GP JASC,PASHR JASC,PSAT,RTKSTAT JASC,PTSS1 JASC,RTCM3
JASC,CMR	New topic
JASC,D1 (RD1)	Added Example section and updated Additional Information section
JASC,DFX	Added Example section and updated Additional Information section
JASC,GL	New topic

JASC,GN	New topic
JASC,GP	New topic
JASC,PASHR	New topic
JASC,PSAT,RTKSTAT	New topic
JASC,PTSS1	New topic
JASC,ROX	Added Example section and updated Additional Information section
JASC,RTCM	Added Example section and updated Additional Information section
JASC,RTCM3	New topic
JASC,VIRTUAL	Added "Example" section
JATT,COGTAU	Corrected "Command Format" section: • Changed `$JTAU,COG,tau<CR><LF>` to `$JATT,COGTAU,cogtau<CR><LF>` • Changed `$JTAU,COG<CR><LF>` to `$JATT,COGTAU<CR><LF>`
JATT,SPDTAU	Corrected "Command Format" section: • Changed `$JTAU,SPEED,tau<CR><LF>` to `$JATT,SPDTAU,spdtau<CR><LF>` • Changed `$JTAU,SPEED<CR><LF>` to `$JATT,SPDTAU<CR><LF>`
JATT,SUMMARY	Combined "Hex Code" table with "Response Components" table in Command Format section and added Example section
JATT,TILTAID	Updated Receiver Response section
JBAUD	Added query information to Command Format section, updated Receiver Response and Additional Information sections, and added Example section
JBIN	Switched first two columns of table in Command Format section, added Bin62, Bin65, Bin66, Bin69, Bin89 messages to table in Command Format section, and added Example section
JBOOT,OMNI	New topic

JCONN	Restructured "connect" command and added query command in Command Format section, and updated Additional Information section
JDIFF	Added "LBand" to table in Command Format section, updated Receiver Response and Additional Information sections, and added Example section
JDIFFX,EXCLUDE	New topic
JDIFFX,GNSSOUT	New topic
JDIFFX,INCLUDE	New topic
JDIFFX,SOURCE	New topic
JDIFFX,TYPE	New topic
JFLASH Overview	New topic
JFLASH,DIR	New topic
JFLASH,FILE,CLOSE	New topic
JFLASH,FILE,NAME	New topic
JFLASH,FILE,OPEN	New topic
JFLASH,FREESPACE	New topic
JFLASH,NOTIFY,CONNECT	New topic
JFLASH,QUERYCONNECT	New topic
JFREQ	Updated Command Format and Receiver Response sections and added Example section
JGEO	Added "[,ALL]" to command in Command Format section, added Example section, and moved text from Additional Information section to Example section
JHP Overview	New topic
JHP,LIMIT	New topic

JHP,MODE,AUTOSEED	New topic
JHP,MODE,IGNORECONV	New topic
JHP,POS	New topic
JHP,POS,LAT,LON,HGT	New topic
JHP,POS,LAT,LON,HGT,,,,OTHER	New topic
JHP,POS,OTHER	New topic
JHP,POS,PRESENT	New topic
JHP,RESET,ACCURACY	New topic
JHP,RESET,ENGINE	New topic
JHP,SEED	New topic
JHP,SEED,LAT,LON,HGT	New topic
JHP,STATIC	New topic
JHP,STATUS,AUTOSEED	New topic
JI	Shortened fields in Receiver Response section format and table and added Example section (moved text from Receiver Response section to Example section)
JK	Moved response text from Command Format section to Receiver Response section and added Example section
JLIMIT	Added query information to Command Format and Receiver Response sections, added Example section, and updated Additional Information section
JLXBEAM	Updated format and table in Receiver Response section (added lonrad, latrad, beamrot,*) and added Example section (moved text from Receiver Response section to Example section)
JMASK	Added Example section and updated Additional Information section
JMODE	New topic

JMODE Overview	New topic
JMODE,FOREST	New topic
JMODE,GPSONLY	New topic
JMODE,L1ONLY	New topic
JMODE,MIXED	New topic
JMODE,NULLNMEA	New topic
JMODE,SBASR	New topic
JMODE,TIMEKEEP	New topic
JMODE,TUNNEL	New topic
JMSG99	New topic
JNMEA,GGAALLGNSS	New topic
JNMEA,PRECISION	New topic
JNP	Added query information to Command Format and Receiver Response sections and updated Additional Information section
JOFF	Added `$JOFF,PORTC<CR><LF>` command to turn off all messages on Port C and updated Additional Information section
JOFF,ALL	New topic
JOMS	Shortened fields in Receiver Response format and table and added Example section
JPOS	Added query information to Command Format and Receiver Response sections
JQUERY,RTKSTAT	New topic
JRAD_Overview	Updated descriptions in table
JRAD,1	Updated Receiver Response section and added Example section

JRAIM	Added query information to Command Format and Receiver Response sections and added Example section
JRELAY	New topic
JRTK Overview	Added JRTK,28 to table
JRTK,1	Updated Receiver Response section and added Example section
JRTK,1,LAT,LON,HEIGHT	Updated Description section
JRTK,1,P	Updated all information in topic
JRTK,12	Added **Warning** at top of topic and updated Description, Receiver Response, and Additional Information sections
JRTK,17	Updated Command Format, Receiver Response, and Additional Information sections and added Example section
JRTK,18	Updated Command Format and Receiver Response sections
JRTK,28	New topic
JRTK,5	Updated Description, Command Format, and Receiver Response sections
JRTK,5,Transmit	Updated Description, Command Format, and Receiver Response sections
JRTK,6	Updated Command Format section and added Example section
JSHOW	Added "PORT" to optional ',SUBSET' data field in Receiver Response section and added Example section
JSMOOTH	Updated Command Format, Receiver Response, and Additional Information sections and added Example section
JT	Updated table in Receiver Response section
JTAU,COG	Added query response and example
JTAU,SPEED	Added query response and example
JWAASPRN	Updated all information in topic

Local Differential and RTK Commands	Added the following commands: JASC,CMR JASC,RTCM3 JASC,PSAT,RTKSTAT JQUERY,RTKSTAT JRTK,28
L-Band Automatic Tracking	Reworded for clarity and added link to JFREQ command
L-Band Commands	Added JBOOT,OMNI command and JHP commands
PASHR	New topic
PCSI,1	Updated Additional Information section to describe different responses depending on whether or not you are connected directly to the SBX-4
PCSI,3	Renamed to PCSI,3,1
PCSI,3,1	Renamed from PCSI,3 command and updated Receiver Response section to include field descriptions
PCSI,3,2	Made the following changes: • Added "beacon" to Description; now reads "Display the ten closest beacon stations" • Removed "time" and "date" from Receiver Response format and table (see updated description for "name" in table) • Expanded definition of "name" in Receiver Response table • Formatted Example to align response components
PCSI,3,3	Updated command description
PSAT,RTKSTAT	New topic
RD1	Revised message component names (for consistency compared to other commands) and descriptions (to provide more information)
Reference Documents	Updated contact information
RTCM SC-104 Protocol	Clarified message support information
SChannelData	New topic (related to Bin89 and Bin99 messages)

SGLONASS_String	New topic (related to Bin62 and Bin65 messages)
SGLONASSChanData	New topic (related to Bin69 message)
SObsPacket	New topic (related to Bin66 message)
SSVAlmanData	New topic (related to Bin98 message)
Subscribing to an Application	Corrected grammatical errors and added link to Hemisphere GPS website
Troubleshooting	Changed from table to list (with drop-down text that appears when you click any item in the list)
TSS1	New topic
Understanding Additive Codes	Updated examples at bottom of topic
Using RightArm to Load Firmware	Updated procedure

Topic Last Updated: v1.02 / January 25, 2011

Change History v1.01

GPGRS	Message Format section: changed format of message components mode and range residuals (also added links in Description column of table)
GPGST	Message Format section: In Message Component column, changed HHMMSS.SSS to HHMMSS.SS to match format of message
GPRMC	Message Format section: Updated decimal minutes portion of latitude and longitude components in format and table so both are five decimal places (latitude = DDMM.MMMMM, longitude = DDDMM.MMMMM)
PCSI,2	Receiver Response and Additional Information sections: Split format of response into two lines and changed Word Error Rate from "Q" to "WER"

Topic Last Updated: v1.01 / September 23, 2010

Change History All Topics Alphabetical

Atlas Commands `	*v.1.07* New topic
Beacon Messages	*v1.02* Removed topic; information now part of Beacon Receiver Commands and Messages
Beacon Receiver Commands and Messages	*v1.02* Added information from now-removed Beacon Messages topic
Bin1	*v1.04* Updated 'VEast' description to say "m/s" and not 'n/s"
Bin16	*v. 1.07* New topic
Bin44	*v.1.07* *New topic*
Bin62	*v1.02* New topic
Bin65	*v1.02* New topic
Bin66	*v1.02* New topic *v1.03* Updated description to refer to GLONASS L1/L2 instead of just GLONASS L1
Bin69	*v1.03* Updated description to refer to GLONASS L1/L2 instead of just GLONASS L1 *v1.02* New topic
Bin76	*v1.03* **Fixed spelling error:** • In Description section under "To determine L1P or L2P", changed "buts" to "bits" in step 1 to read "Use the lower 16 bits provided in the message." • In Message format section corrected spelling errors:(1) changed "port" to "part" for Carrier Phase (High part) in both instances of P7_Doppler_FL row, and (2) changed "Cide" to "Code" in both instances of CodeAndPhase row

Bin89	*v1.02* New topic
Bin94	v1.04 Updated description for 'r' value in Command Format to Request Message

	section to the following: 'r' = 1 (on) or 0 (off) When set to on the message is sent once and then sent again whenever satellite information changes
Bin95	v1.04 Updated description for 'r' value in Command Format to Request Message section to the following: 'r' = 1 (on) or 0 (off) When set to on the message is sent once (one message for each tracked satellite at 1 second intervals) and then sent again whenever satellite information changes
Bin98	*v1.03* Added "GPS" to description *v1.02* AlmanData[8] is now a single row in the Message Format table with a link to SSVAlmanData that provides detailed information
Bin99	*v1.02* sChannelData[CHANNELS_12] is now a single row in the Message Format table with a link to SChannelData that provides detailed information
Binary Messages	*v1.03* Updated descriptions for the following in the message table: • Bin66 - changed GLONASS L1 to GLONASS L1/L2 • Bin69 - changed GLONASS L1 to GLONASS L1/L2 • Bin98 - added "GPS" to description *v1.02* Added information on new messages (Bin62, Bin65, Bin66, Bin69, Bin89)
Binary Messages Code	*v1.02* Updated to current version of code
COAST Technology	*v1.02* Added **Note** regarding Crescent and Eclipse boards
Commands and Messages Overview	*v1.02* Added link for GLONASS Commands and Messages and removed link for "Beacon Messages" since that information is now part of Beacon Receiver Commands and Messages
Configuring the Data Message Output	*v1.02* Added **Note** regarding specifying Port T when writing to a USB flash drive
Crescent Vector Commands and Messages	*v1.02* Added PASHR and TSS1 to Crescent Vector messages table

Data Messages	*v1.04* Added the PSAT,RTKPROG message *v1.02* Added the GPALM message
Determining the Receiver Type and Current Application	*v1.02* Updated both tables
DGPS Base Station Commands	*v1.02* Updated definitions of commands
e-Dif Commands	*v1.02* Updated definitions of commands
Eclipse II Subscription Codes	*v1.03* **Fixed spelling errors:** • Removed redundant column on far right of table • Changed "eDiff" to "e-Dif" • Changed "Raw Ou" to "Raw Out"
Ethernet Configuration	*v1.07* New topic
EGNOS	*v1.02* Changed reference to "three satellites" to now read "multiple geostationary satellites and a network of ground stations"
Galileo Commands	*v1.07* *Added note:* *Note: For observations in tracking status, see GNSS, Bin 16 & Bin 19. Added: **"**Commands and Messages"
General Operation and Configuration Commands	*v1.04* Added the following commands: • JDIFF,AVAILABLE • JFORCEAPP • JMODE,BASE • JMODE,FIXLOC • JMODE,GLOFIX • JMODE,SBASNORTK • JMODE,SURETRACK

- JPRN,EXCLUDE
- JSHOW,ASC
- JSHOW,BIN

- JSHOW,CONF
- JSHOW,GP

v1.03

Added JSHOW,THISPORT command

v1.02

Added the following commands to the table:

- JDIFFX,EXCLUDE
- JDIFFX,GNSSOUT
- JDIFFX,INCLUDE
- JDIFFX,SOURCE
- JDIFFX,TYPE
- JFLASH,DIR
- JFLASH,FILE,CLOSE
- JFLASH,FILE,NAME
- JFLASH,FILE,OPEN
- JFLASH,FREESPACE
- JFLASH,NOTIFY,CONNECT
- JFLASH,QUERYCONNECT
- JMODE
- JMODE,FOREST
- JMODE,GPSONLY
- JMODE,L1ONLY
- JMODE,MIXED
- JMODE,NULLNMEA
- JMODE,SBASR
- JMODE,TIMEKEEP
- JMODE,TUNNEL
- JRELAY

GLMLA	v1.04 Removed 'JASC' from beginning of response (just after '$') in Message Format section

	v1.02 New topic
GLONASS Commands and Messages	v1.02 New topic
GNGSA	*v1.07* Document system ID 6
GPALM	*v1.04* Updated Message format section *v1.02* New topic
GPCRQ,MSK	*v1.02* Added Example section
GPCRQ,MSS	*v1.02* Added Example section
GPDTM	*v1.04* Updated Message format table for consistency
GPGGA	*v1.04* Updated Message format table for consistency v1.02 Added Note at top of topic, updated Command Format to Request Message section to add "5" and "4" to list of message rates, added Example section, and updated Additional Information and Related Commandssections
GPGLL	*v1.02* Added Note at top of topic and updated Additional Information and Related Commands sections
GPGNS	*v1.04* Updated Message format table for consistency and added "NS" field (navigational status) *v1.03* Updated description of "mm" field (Mode indicator) in Message Format section *v1.02* Added Note at top of topic and updated Additional Information and Related Commands sections
GPGRS	*v1.04* Added "GSID" field (GNSS system ID) and "SID" field (signal ID) to Message Format section; also changed max output rate to 1 (so in the Command Format to Request Message section, instead of "20, 10, 2, 1, 0 or .2" it now says "1, 0 or .2") *v1.01* Message Format section: changed format of message components mode and range residuals (also added links in Description column of table)

GPGSA	*v1.07* Changed to “GNGSA” (where applicable) *v1.04* Added "GSID" field (GNSS system ID) to Message Format section *v1.02* Added Note at top of topic and updated Additional Information and Related Commands sections
GPGST	*v1.01* Message Format section: In Message Component column, changed HHMMSS.SSS to HHMMSS.SS to match format of message
GPGSV	*v1.04* Added "SID" field (signal ID) to Message Format section *v1.02* Added Note at top of topic and updated Additional Information and Related Commands sections
GPHDM	*v1.02* Corrected alternate format in Message Format section from $H**C**HDM,x.x,M*CC<CR><LF> to $H**E**HDM,x.x,M*CC<CR><LF>
GPMSK	*v1.02* Updated Receiver Response section and added Example section
GQGSV	*v1.07* Section added
GPRMC	*v1.04* Added "M" field (mode indicator) and "NS" field (navigational status) to Message Format section *v1.01* Message Format section: Updated decimal minutes portion of latitude and longitude components in format and table so both are five decimal places (latitude = DDMM.MMMMM, longitude = DDDMM.MMMMM)
Hardware Platforms Overview	*v1.03* New topic
Interpreting the $JI and $JK 'Date'/Subscription Codes	*v1.03* Changed shading at bottom of topic to only shade '3000', not'01/01/3000'
JAGE	*v1.02* Added Example section and updated Additional Information section
JAIR	*v1.02* Added query code to Command Format section and added Example section
JALT	*v1.03* Updated Command Format, Receiver Response, and Example sections to more clearly define 'h' value

	v1.02 Added "SATS" as a "c" value in the Command Format section, added Example section, and updated Additional Information section
JAPP	*v1.02* Categorized Command Format section by receiver type, moved receiver response text from Command Format section to Receiver Response section, and added Example section
JASC Overview	*v1.07* Removed: JASC, PSAT,SMARTBASE Added: JASC< PSAT< VCT,1 JASC, GG Command section added Replaced JASC, GP to JASC GN *v1.02* Added the following messages to the table: JASC,CMR JASC,GL JASC,GN JASC,GP JASC,PASHR JASC,PSAT,RTKSTAT JASC,PTSS1 JASC,RTCM3
JASC,CMR	*v1.02* New topic
JASC,D1 (RD1)	*v1.02* Added Example section and updated Additional Information section
JASC,DFX	*v1.02* Added Example section and updated Additional Information section
JASC,GL	*v1.02* New topic
JASC,GN	*v1.04* Corrected "MSG" column entries to begin with "GN" instead of "GP" *v1.02* New topic
JASC,GP	*v1.04* Changed max output rate for GPGRS to 1 (so in the Command Format to Request Message section, instead of "20, 10, 2, 1, 0 or .2" it now says "1, 0 or .2") *v1.02* New topic

JASC,PASHR	*v1.02* New topic
JASC,PSAT,RTKPROG	*v1.04*

	New topic
JASC,PSAT,RTKSTAT	*v1.04* In Additional Information section removed incorrect text stating "To query the receiver for the current setting, issue the JSHOW command." *v1.02* New topic
JASC,PTSS1	*v1.03* Removed checksum; added units for Heave, Pitch, and Roll; changed "gyrocompass settle time" in 'h' description from "several hours" to "approximately five minutes" *v1.02* New topic
JASC,ROX	*v1.02* Added Example section and updated Additional Information section
JASC,RTCM	*v1.02* Added Example section and updated Additional Information section
JASC,RTCM3	*v1.02* New topic
JASC,VIRTUAL	*v1.02* Added "Example" section
JATT,COGTAU	*v1.04* Added following paragraph in Description section: "COG is computed using only the primary GPS antenna (when using a multi-antenna system) and its accuracy depends upon the speed of the vessel (noise is proportional to 1/speed). This value is invalid when the vessel is stationary, as tiny movements due to calculation inaccuracies are not representative of a vessel's movement." *v1.02* Corrected "Command Format" section: • Changed `$JTAU,COG,tau<CR><LF>` to `$JATT,COGTAU,cogtau<CR><LF>` • Changed `$JTAU,COG<CR><LF>` to `$JATT,COGTAU<CR><LF>`
JATT,HRTAU	*v1.04* Changed heading rate time constant to rate of turn (ROT) time constant
JATT,HTAU	*v1.04* Updated Description section
JATT,PBIAS	*v1.04*

In Additional Information section added text after first sentence

JATT,PTAU	*v1.04* Updated Description section
JATT,SPDTAU	*v1.02* Corrected "Command Format" section: • Changed `$JTAU,SPEED,tau<CR><LF>` to `$JATT,SPDTAU,spdtau<CR><LF>` • Changed `$JTAU,SPEED<CR><LF>` to `$JATT,SPDTAU<CR><LF>`
JATT,SUMMARY	*v1.02* Combined "Hex Code" table with "Response Components" table in Command Format section and added Example section
JATT,TILTAID	*v1.02* Updated Receiver Response section
JBAUD	*v1.02* Added query information to Command Format section, updated Receiver Response and Additional Information sections, and added Example section
JBIN	*v1.07* Added Bin16, Bin44 *v1.03* Updated descriptions for the following in the message table: • Bin66 - changed GLONASS L1 to GLONASS L1/L2 • Bin69 - changed GLONASS L1 to GLONASS L1/L2 • Bin98 - added "GPS" to description *v1.02* Switched first two columns of table in Command Format section, added Bin62, Bin65, Bin66, Bin69, Bin89 messages to table in Command Format section, and added Example section
JBOOT,OMNI	*v1.07* Removed: "Omni", Added "L-Band". Removed from description "It also allows you to reset the L-band high precision services resolution algorithm." *v1.02* New topic
JCONN	*v1.02* Restructured "connect" command and added query command in Command Format section, and updated Additional Information section
JDIFF	*v1.04* Updated Receiver Response section to show SOURCE and TYPE *v1.02* Added "LBand" to table in Command Format section, updated Receiver

Response and Additional Information sections, and added Example section

JDIFF,AVAILABLE	*v1.04* New topic
JDIFFX,EXCLUDE	*v1.04* Added information for querying current setting *v1.02* New topic
JDIFFX,GNSSOUT	*v1.07* Added: BEIDOU, GALILEO to Command Format. Replaced “both” GPOS and GLONASS with “all”GPS and GLONASS *v1.04* Throughout topic replaced "GNSS output in correction formats" with "GNSS systems to be output in the differential" *v1.02* New topic
JDIFFX,INCLUDE	*v1.07* Added: [, ATLAS] to Command Format *v1.04* Added information for querying current setting *v1.02* New topic
JDIFFX,SOURCE	*v1.02* New topic
JDIFFX,TYPE	*v1.04* Updated 'type' options in Receiver Response section *v1.03* Corrected Receiver Response from `$>JDIFF,type` to `$>JDIFFX,TYPE,type` and added 'type' list *v1.02* New topic
JETHERNET	*v1.07* *New topic*
JETHERNET MODE	*v1.07* *New topic*
JETHERNET PORT1	*v1.07* *New topic*
JFLASH Overview	*v1.02* New topic
JFLASH,DIR	*v1.02* New topic

JFLASH,FILE,CLOSE	*v1.02* New topic
JFLASH,FILE,NAME	*v1.02* New topic
JFLASH,FILE,OPEN	*v1.02* New topic
JFLASH,FREESPACE	*v1.02*

	New topic
JFLASH,NOTIFY,CONNECT	*v1.02* New topic
JFLASH,QUERYCONNECT	*v1.02* New topic
JFORCEAPP	*v1.04* New topic
JFREQ	*v1.04* UpdatedAtlas satellite table *v1.02* Updated Command Format and Receiver Response sections and added Example section
JGEO	*v1.02* Added "[,ALL]" to command in Command Format section, added Example section, and moved text from Additional Information section to Example section
JHP Overview	*v1.07* Removed topic *v1.02* New topic
JHP,LIMIT	*v1.07* Removed topic *v1.02* New topic
JHP,MODE,AUTOSEED	*v1.07* Removed topic *v1.03* Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure." *v1.02* New topic
JHP,MODE,IGNORECONV	*v1.07* Removed topic *v1.03* Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure." *v1.02* New topic

JHP,POS	*v1.07* Removed topic *v1.02* New topic
JHP,POS,LAT,LON,HGT	*v1.07* Removed topic *v1.02* New topic

JHP,POS,LAT,LON,HGT,,,,OTHER	*v1.07* Removed topic *v1.03* Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure." *v1.02* New topic
JHP,POS,OTHER	*v1.07* Removed topic *v1.03* Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure." *v1.02* New topic
JHP,POS,PRESENT	*v1.07* Removed topic *v1.02* New topic
JHP,RESET,ACCURACY	*v1.07* Removed topic *v1.02* New topic
JHP,RESET,ENGINE	*v1.07* Removed topic *v1.02* New topic
JHP,SEED	*v1.07* Removed topic *v1.02* New topic
JHP,SEED,LAT,LON,HGT	*v1.07* Removed topic *v1.02* New topic
JHP,STATIC	*v1.07* Removed topic *v1.02* New topic
JHP,STATUS,AUTOSEED	*v1.07* Removed topic

	v1.03 • Added the following Note at top of topic: "The autoseeding function is available on the Eclipse II and miniEclipse with LX2 platforms. Autoseed is not compatible with the original Eclipse's memory structure." • Added definitions for 'status' field in Receiver Response section *v1.02* New topic
JI	*v1.02* Shortened fields in Receiver Response section format and table and added Example section (moved text from Receiver Response section to Example section)

JK	*v1.03* Added 'DowngradeCode' field to Receiver Response section and updated response descriptions and Example section accordingly *v1.02* Moved response text from Command Format section to Receiver Response section and added Example section
JLIMIT	*v1.02* Added query information to Command Format and Receiver Response sections, added Example section, and updated Additional Information section
JLXBEAM	*v1.07* Added:"Receiver Response" commands *v1.02* Updated format and table in Receiver Response section (added lonrad, latrad, beamrot,*) and added Example section (moved text from Receiver Response section to Example section)
JMASK	*v1.02* Added Example section and updated Additional Information section
JMODE	*v1.04* Added receiver responses for BASE, FIXLOC, GLOFIX, SBASNORTK, and SURETRACK *v1.02* New topic
JMODE Overview	*v1.04* Added the following commands: • JMODE,BASE • JMODE,FIXLOC • JMODE,GLOFIX • JMODE,SBASNORTK • JMODE,SURETRACK *v1.02* New topic
JMODE,BASE	*v1.04* New topic
JMODE,FIXLOC	*v1.04* New topic
JMODE,FOREST	*v1.02* New topic

JMODE,GLOFIX	*v1.04* *New topic*
JMODE,GPSONLY	*v1.02* New topic
JMODE,L1ONLY	*v1.02* New topic
JMODE,MIXED	*v1.04* Corrected query responses: • $>JMODE,MIXED,**ON** changed to $>JMODE,MIXED,**YES** • $>JMODE,MIXED,**OFF** changed to $>JMODE,MIXED,**NO** *v1.02* New topic
JMODE,NULLNMEA	*v1.03* Corrected responses (in Receiver Response and Example sections): • Changed `$>JMODE,NULLNMEA,ON` to `$>JMODE,NULLNMEA,YES` • Changed `$>JMODE,NULLNMEA,OFF` to `$>JMODE,NULLNMEA,NO` *v1.02* New topic
JMODE,SBASNORTK	*v1.04* New topic
JMODE,SBASR	*v1.04* Corrected query responses: • $>JMODE,SBASR,**ON** changed to **$>JMODE,SBASR,**YES • $>JMODE,SBASR,**OFF** changed to **$>JMODE,SBASR,**NO *v1.02* New topic
JMODE,SURETRACK	*v1.04* New topic
JMODE,TIMEKEEP	*v1.04* Corrected query responses:

	• $>JMODE,TIMEKEEP,**ON** changed to **$>JMODE,TIMEKEEP,**YES • $>JMODE,TIMEKEEP,**OFF** changed to **$>JMODE,TIMEKEEP,**NO *v1.02* New topic
JMODE,TUNNEL	*v1.04* Corrected query responses: • $>JMODE,TIMEKEEP,**ON** changed to **$>JMODE,TIMEKEEP,**YES • $>JMODE,TIMEKEEP,**OFF** changed to **$>JMODE,TIMEKEEP,**NO *v1.02* New topic
JMSG99	*v1.02* New topic
JNMEA,GGAALLGNSS	*v1.02* New topic
JNMEA,PRECISION	*v1.04* Added GPGNS to list of messages (in Description section) for which you can set the decimal places output *v1.02* New topic
JNP	*v1.04* Added GPGNS to list of messages (in Description section) for which you can set the decimal places output *v1.02* Added query information to Command Format and Receiver Response sections and updated Additional Information section
JOFF	*v1.02* Added `$JOFF,PORTC<CR><LF>` command to turn off all messages on Port C and updated Additional Information section
JOFF,ALL	*v1.02* New topic
JOMS	*v1.07* Removed topic *v1.02* Shortened fields in Receiver Response format and table and added Example section

JPOS	*v1.02* Added query information to Command Format and Receiver Response sections
JPRN,EXCLUDE	*v1.07* Added: : 'z,z,z…' represents the GALILEO PRNs you want to exclude", "Exclude no GALILEO PRNs: $JPRN,EXCLUDE,GAL,NONE<CR><LF>" *v1.04* New topic
JQUERY,RTKPROG	*v1.04* New topic
JQUERY,RTKSTAT	*v1.02* New topic
JRAD_Overview	*v1.02* Updated descriptions in table
JRAD,1	*v1.02* Updated Receiver Response section and added Example section
JRAD,7	*v1.04* Updated Receiver Response from `$>` to `$>JRAD,7,OK`
JRAD,9	*v1.04* New name of previous JRAD,9,1,1 command. Added information on "JRAD,9,0" that turns base mode off
JRAD,9,1,1	*v1.04* Changed command name to JRAD,9
JRAIM	*v1.02* Added query information to Command Format and Receiver Response sections and added Example section
JRELAY	*v1.02* New topic
JRTK Overview	*v1.03* Added JRTK,18,BEARING and JRTK,18,NEU *v1.02* Added JRTK,28
JRTK,1	*v1.02* Updated Receiver Response section and added Example section

JRTK,1,LAT,LON,HEIGHT	*v1.02*

	Updated Description section
JRTK,1,P	*v1.02* Updated all information in topic
JRTK,12	*v1.02* Added **Warning** at top of topic and updated Description, Receiver Response, and Additional Information sections
JRTK,17	*v1.02* Updated Command Format, Receiver Response, and Additional Information sections and added Example section
JRTK,18	*v1.02* Updated Command Format and Receiver Response sections
JRTK,18,BEARING	*v1.03* New topic
JRTK,18,NEU	*v1.03* New topic
JRTK,28	*v1.02* New topic
JRTK,5	*v1.02* Updated Description, Command Format, and Receiver Response sections
JRTK,5,Transmit	*v1.02* Updated Description, Command Format, and Receiver Response sections
JRTK,6	*v1.02* Updated Command Format section and added Example section
JSHOW	*v1.03* Updated JFREQ (line 15 in table) to add information on bit rate and AUTO *v1.02* Added "PORT" to optional ',SUBSET' data field in Receiver Response section and added Example section
JSHOW,ASC	*v1.04* New topic
JSHOW,BIN	*v1.04* New topic

JSHOW,CONF	*v1.04* New topic (some of the information in this topic appeared in the previous JSHOW topic)
JSHOW,GP	*v1.04* New topic (some of the information in this topic appeared in the previous JSHOW topic)
JSHOW,THISPORT	*v1.03* New topic
JSIGNAL Command	1.*07* New topic Description rewritten to read: “Set the receiver to use the specify signal: GNSS signals that the receiver will attempt to track. Specific signals shown here are only valid for receivers supporting the signal in question.” Added: •$>JSIGNAL,INCLUDE[,L1CA][,L1P][,L2P][,L2C][,G1][,G2][,E1BC][,B1][,B2][,B3] [,E5B][,QZSL1CA][,QZSL2C]<CR><LF> Changed Command Format to read: •Specify the signal(s) to be used $JSIGNAL,INCLUDE[,L1CA][,L1P][,L2P][,L2C][,G1][,G2][,E1BC][,B1][,B2][,B3] [,E5B][,QZSL1CA][,QZSL2C][,ALL]<CR><LF> Specify the signal(s) NOT to be used $JSIGNAL,EXCLUDE[,L1CA][,L1P][,L2P][,L2C][,G1][,G2][,E1BC][,B1][,B2][,B3] [,E5B][,QZSL1CA][,QZSL2C][,ALL]<CR><LF>
JSMOOTH	*v1.04* Added 'DEFAULT' to Command Format section and moved response text (regarding SHORT and LONG) from Command Format section to Receiver Response section *v1.02* Updated Command Format, Receiver Response, and Additional Information sections and added Example section
JT	*v1.03* Added Receiver Response information for miniEclipse *v1.02* Updated table in Receiver Response section
JTAU,COG	*v1.02* Added query response and example
JTAU,SPEED	*v1.02* Added query response and example
JWAASPRN	*v1.02* Updated all information in topic
Local Differential and RTK Commands	*v1.04* Added the following commands and message: • JASC,PSAT,RTKPROG • JQUERY,RTKPROG

- PSAT,RTKPROG

v1.03

Added the following commands:

- JRTK,18,BEARING
- JRTK,18,NEU

v1.02

Added the following commands:

- JASC,CMR
- JASC,RTCM3
- JASC,PSAT,RTKSTAT
- JQUERY,RTKSTAT
- JRTK,28

L-Band Automatic Tracking	*v1.07* Removed: high-precision, and high precision with GLONASS services" 4.L-band (DGPS) Replaced: "DGPS" with "Atlas *v1.02* Reworded for clarity and added link to JFREQ command
*L-B*and Commands	*v1.02* Added JBOOT,OMNI command and JHP commands
NMEA 0183 Message	*v1.07* *XX NMEA 0183 talker field (GP = GPS, GL = GLONASS, GA = GALILEO, GB = BEIDOU, GN = All constellations)*
PASHR	*v1.02* New topic
PCSI,0	*v1.04* Changed Command Type to link to Beacon Receiver topic
PCSI,1	*v1.04* Changed Command Type to link to Beacon Receiver topic *v1.02* Updated Additional Information section to describe different responses depending on whether or not you are connected directly to the SBX-4
PCSI,1,1	*v1.04* Moved example from Receiver Response section to new Example section
PCSI,2	*v1.04* Added Example section and changed Command Type to link to Beacon Receiver topic *v1.01* Receiver Response and Additional Information sections: Split format of response into two lines and changed Word Error Rate from "Q" to "WER"
PCSI,3	*v1.02* Renamed to PCSI,3,1
PCSI,3,1	*v1.04* Added Example section and changed Command Type to link to Beacon Receiver topic *v1.02*

Renamed from PCSI,3 command and updated Receiver Response section to include field descriptions

PCSI,3,2	*v1.04* Added `$PCSI,ACK,3,2` as first line of receiver response *v1.03* Added 'time' and 'date' back to Receiver Response format and table v*1.02* Made the following changes: • Added 'beacon' to Description; now reads "Display the ten closest beacon stations" • Removed 'time' and 'date' from Receiver Response format and table (see updated description for "name" in table) • Expanded definition of 'name' in Receiver Response table • Formatted Example to align response components
PCSI,3,3	*v1.04* Added `$PCSI,ACK,3,3` as first line of receiver response *v1.02* Updated command description
PCSI,4	*v1.04* New topic
PCSI,5	*v1.04* New topic
PCSI,6	*v1.04* New topic
PCSI,7	*v1.04* New topic
Post-Processing	*v1.07* *Added:* *the following messages, which must be logged in a binary file:* *Observations: Bin 76 (GPS), Bin 66 (GLONASS), Bin 36 (BEIDOU)* *Or* *Bin 16 (All constellations; required for GALILEO)* *Ephemeris: Bin 95 (GPS), Bin 65 (GLONASS), Bin 35 (BEIDOU), Bin 45 (GALILEO)* *Time conversion: Bin 94 (GPS), Bin 34 (BEIDOU), Bin 44 (GALILEO)* *Changed:* *(Crescent receivers must log Bin 94, 95, and 96 messages for GPS). Depending on the application, the binary data can be logged to a file and then translated to RINEX at a later time on a PC.*
PSAT,GBS	*v1.04* Added "GSID" field (GNSS system ID) and "SID" field (signal ID) to Message Format section
PSAT,RTKPROG	*v1.04* New topic

PSAT,RTKSTAT	*v1.04* Updated "ACCSTAT" field (accuracy status), added "SNT" field (ionospheric scintillation) field, and removed CMR+ from TYP (will show as CMR) in Message Format section; also added text regarding JSAVE in Additional Information section *v1.03* Added description for ACCSTAT (accuracy status) parameter in message response

	v1.02 New topic
Quick Start	*v1.03* New topic
QZSS Commands and Messages	*v1.07* *Section added*
RD1	*v1.03* Removed checksum (*CC) from message format *v1.02* Revised message component names (for consistency compared to other commands) and descriptions (to provide more information)
Reference Documents	*v1.02* Updated contact information
RTCM SC-104 Protocol	*v1.02* Clarified message support information
SChannelData	*v1.02* New topic (related to Bin89 and Bin99 messages)
SGLONASS_String	*v1.02* New topic (related to Bin62 and Bin65 messages)
SGLONASSChanData	*v1.02* New topic (related to Bin69 message)
SObsPacket	*v1.02* New topic (related to Bin66 message)
SSVAlmanData	*v1.02* New topic (related to Bin98 message)
Subscribing to an Application	*v1.02* Corrected grammatical errors and added link to Hemisphere GPS website
Troubleshooting	*v1.02* Changed from table to list (with drop-down text that appears when you click any item in the list)
TSS1	*v1.03* Removed checksum; added units for Heave, Pitch, and Roll; changed "gyrocompass settle time" in 'h' description from "several hours" to "approximately five minutes" *v1.02* New topic

Understanding Additive Codes	*v1.03* Updated examples at bottom of topic *v1.02* Added link for Eclipse II codes, added Table 3 (additive code components), and moved example text to end of topic
Universal Development Kit	*v1.03* New topic
Using RightArm to Load Firmware	*v1.07* Re-numbered list for accuracy *v1.02* Updated procedure

Topic Last Updated: v1.07 Feburary 14, 2017

Troubleshooting

Use the following checklist to troubleshoot anomalous receiver system operation.

Receiver fails to power No

data from receiver

Random data from receiver No

GPS lock

No SBAS lock

No DGPS position in external RTCM mode

Non-differential GPS output

Multipath signals

Intermittent GPS Lock

Topic Last Updated: v1.02 / January 25, 2011