The GPS Toolkit

A User's Guide for Scientists, Engineers and Students

Tracie Conn, Tom Gaussiran, R. Benjamin Harris, Jon Little
Richard Mach, David Munton, Brent Renfro, Brian Tolman
Timothy Craddock

Applied Research Laboratories, The University of Texas at Austin

Martin Vermeer

Department of Surveying, Helsinki University of Technology

August 24, 2012
GPSTk Revision 1038

Contents

The goal of the GPSTk project is to provide a world class, open source computing suite to the satellite navigation community. It is our hope that the GPSTk will empower its users to perform new research and to create new applications.

GPS users employ practically every computational architecture and operating system. Therefore the design of the GPSTk suite is as platform-independent as possible. Platform independence is achieved through use of the ANSI-standard C++ programming language. The principles of object-oriented programming are used throughout the GPSTk code base in order to ensure that the code is modular, extensible, and maintainable.

The GPSTk suite consists of a core library, auxiliary libraries, and a set of applications. The core library provides a wide array of functions that solve processing problems associated with GPS such as processing or using RINEX. The library is the basis for the more advanced applications distributed as part of the GPSTk suite.

The GPSTk is sponsored by Space and Geophysics Laboratory, within the Applied Research Laboratories at the University of Texas at Austin (ARL:UT). GPSTk is the by-product of GPS research conducted at ARL:UT since before the first satellite launched in 1978; it is the combined effort of many software engineers and scientists. In 2003 the research staff at ARL:UT decided to open source much of their basic GPS processing software as the GPSTk.

Part I

Theory

Chapter 1

The Global Positioning System in a Nutshell

The Global Positioning System is actually a U.S. government satellite navigation system that provides a civilian signal. As of this writing, the signal is broadcast simultaneously by a constellation of 32 satellites each with a 12 hour orbit. From any given position on the Earth, 8 to 12 satellites are usually visible at a time.

1.1 GPS in a Nutshell

Each satellite broadcasts spread spectrum signals at 1575.42 and 1227.6 MHz, also known as L1 and L2, respectively. Currently the civil signal is broadcast only on L1. The signal contains two components: a time code and a navigation message. By differencing the received time code with an internal time code, the receiver can determine the distance, or range, that the signal has traveled. This range observation is offset by errors in the (imperfect) receiver clock; therefore it is called a pseudorange. The navigation message contains the satellite ephemeris, which is a numerical model of the satellite's orbit.

GPS receivers record, besides the pseudorange, a measurement called the carrier phase (or just phase); it is also a range observation like the pseudorange, except (1) it has an unknown constant added to it (the phase ambiguity) and (2) it is much smoother (about 100 times less measurement noise than the pseudorange!), which makes it useful for precise positioning. Because of the way it is measured, the phase is subject to random, sudden jumps; these discrete changes always come in multiples of the wavelength of the GPS signal, and are called cycle slips.

1.1.1 The Position Solution

The standard solution for the user location requires a pseudorange measurement and an ephemeris for each satellite in view. At least four measurements are required as there are four unknowns: 3 coordinates of position plus the receiver clock offset. The basic algorithm for the solution is described in the official GPS Interface Control Document, or ICD-GPS-200.

The position solution is corrupted due to two sources of error: errors in the observations and errors in the ephemeris.

Reducing Measurement Errors

The GPS signal travels through every layer of the Earth's atmosphere. Each layer affects the signal differently. The ionosphere, which is the high-altitude, electrically charged part of the atmosphere, introduces a delay, and therefore a range error, into the signal. The ionosphere delay can be predicted using a model. However, the accuracy of ionosphere models is limited. A better alternative is to measure and remove the ionosphere delay. Measurement of the ionosphere delay is possible by taking advantage of the fact that the delay is frequency dependent. It can be directly computed if you have data on both the GPS frequencies. There is also a delay due to the troposphere, the lower part of the atmosphere. Like the ionosphere delay, the atmosphere delay can be either predicted or derived from measurements. There are many other errors associated with the GPS signal: multipath reflections and relativistic effects are two examples.

More precise applications reduce the effect of error sources by a technique referred to as differential GPS (DGPS). By differencing measurements simultaneously collected by the user and a nearby reference receiver, the errors that are common to both receivers (most of them) are removed. The result of DGPS positioning is a position relative to the reference receiver; adding the reference position to the DGPS solution results in the absolute user position.

The alternative to DGPS is to explicitly model and remove errors. Creating new and robust models of phenomena that affect the GPS signal is an area of active research at ARL:UT and other laboratories. The positioning algorithm can be used to explore such models. Essentially, the basic approach is to turn the positioning algorithm inside out to look at the corrections themselves. For example, observations from a network of receivers can create a global map or model of the ionosphere.

Improved Ephemerides

The GPS position solution can be directly improved by using an improved satellite ephemeris. The U.S National Geospatial-Intelligence Agency (NGA) generates and makes publicly available a number of precise ephemerides, which are more accurate satellite orbits [5], [3]. Satellite orbits described by the broadcast navigation message have an error on the order of meters; the precise ephemeris has decimeter accuracy. The International GNSS Service (IGS) is a global, civil cooperative effort that also provides free precise ephemeris products [4]. Global networks of tracking stations produce the observations that make generation of the precise ephemerides possible.

1.2 GPS Data Sources

GPS observation data from many tracking stations are freely available on the Internet. Many such stations contribute their data to the IGS. In addition, many networks of stations also post their data to the Internet; for example the Australian Regional GPS Network (ARGN) [1] and global cooperatives such as NASA's Crust Dynamics Data Information System (CDDIS) [2].

1.2.1 GPS File Formats

Typically GPS observations are recorded in a standardized format developed by and for researchers. Fundamental to this format is the idea that the data should be independent of the type of receiver that collected it. For this reason the format is called Receiver INdependent Exchange, or RINEX. Another format associated with GPS is SP-3, which records the precise ephemeris. The GPSTk supports both RINEX and SP-3 formats.

1.2.2 Receiver Protocols

GPS receivers have become less expensive and more capable over the years, in particular handheld and mobile GPS receivers. The receivers have many features in common. All of the receivers output a position solution every few seconds. All receivers store a list of positions, called waypoints. Many can display maps that can be uploaded. Many can communicate with a PC or handheld to store information or provide position estimates to plotting software.

Typically communication with a PC and other systems follows a standard provided by the National Marine Electronics Association called NMEA-0183. NMEA-0183 defines an ASCII based format for communication of position solutions, waypoints and a variety of receiver diagnostics. Here is an example of a line of NMEA data, or sentence:

```
$GPGLL,5133.81,N,00042.25,W*75
```

The data here is a latitude, longitude fix at 51 deg 33.81 min North, 0 deg 42.25 min West; the last part is a checksum.

As a public standard, the NMEA-0183 format has given the user of GPS freedom of choice. NMEA-0183 is the format most typically used by open source applications that utilize receiver-generated positions.

Closed standards are also common. SiRF is a proprietary protocol that is licensed to receiver manufacturers. Many receiver manufacturers implement their own binary protocols. While some of these protocols have been opened to the public, some have been reverse engineered.

1.3 References

[1] *Australian Regional GPS Network*. http://www.ga.gov.au/geodesy/argn/.

[2] *Crustal Dynamics Data Information System, NASA's Archive of Space Geodesy Data*. http://cddis.nasa.gov/.

[3] *National Geospatial-Intelligence Agency GEOINT Sciences Office, Global Positioning System (GPS) Division*. http://earth-info.nga.mil/GandG/sathtml/.

[4] G. Beutler, M. Rothacher, S. Schaer, T.A. Springer, J. Kouba, and R.E. Neilan. The International GPS Service (IGS): An Interdisciplinary Service in Support of Earth Sciences. *Advances in Space Research*, 23(4):631–635, 1999.

[5] R. Benjamin Harris, Brian Tolman, Tom Gaussiran, David Munton, Jon Little, Richard Mach, Scot Nelsen, and Brent Renfro. The GPS Toolkit: Open Source GPS Software. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, California, September 2004.

Chapter 2

GPS File Formats

A variety of file formats are supported within the GPSTk. The file formats generally store GPS observation data or data related to processing of GPS observables. In this section, a summary of the file formats supported within the GPSTk is presented along with a brief rationale of why each format is supported within the GPSTk and where to find additional information on the format.

2.1 RINEX

The Receiver INdependent EXchange (RINEX) format was developed by the National Geodetic Survey (NGS) in the U.S. and the University of Berne in Switzerland. RINEX is actually three format definitions that allow storage of GPS observations, GPS navigation message information, and meteorological data associated with GPS observations. GPSTk contains classes to both read and write RINEX V2.1 and V3 data files of all types (observation, navigation message, and meteorological). RINEX has undergone a number of revisions since its inception. Each revision is defined using a standard [5], [2], [3], [4].

2.2 FIC

The Floating, Integer, Character (FIC) format was developed in the mid-80s as a relatively machine-independent way to store GPS observation and navigation message data while retaining receiver specific characteristics. Over time, the RINEX format (see above) proved more popular with users and use of the observation records within the FIC format faded away. However, the FIC records associated with GPS navigation message data are still supported within the GPSTk because these records retain some data quantities that are not contained within the RINEX navigation message file. For example, RINEX makes few provisions for storing the almanac data contained in Subframe 4 and Subframe 5. Like RINEX, a standards document defines FIC [7].

2.3 SP-3

The SP-3 format stores ephemeris information for satellites. Usually SP-3 is used for storage of GPS precise ephemerides. GPSTk supports both SP-3a and SP3-c formats. SP-3 was originally designed by NGS. Standards documents describe the specific details of the SP-3 formats [1], [6].

2.4 References

[1] *The NGS GPS Orbital Formats.*

[2] Werner Gürtner. *RINEX: The Reciever Independant Exchange Format Version 2.10.* http://www.ngs.noaa.gov/CORS/Rinex2.html, 1993.

[3] Werner Gürtner and Lou Estey. *RINEX: The Reciever Independant Exchange Format Version 2.11.* ftp://igscb.jpl.nasa.gov/igscb/data/format/rinex211.txt, 2006.

[4] Werner Gürtner and Lou Estey. *RINEX: The Reciever Independant Exchange Format Version 3.00.* ftp://igscb.jpl.nasa.gov/igscb/data/format/rinex300.pdf, 2006.

[5] Werner Gürtner and Gerald M. Mader. *The RINEX Format: Current Status, Future Developments.* http://navcenter.org/ftp/GPS/REPORTS/rinex.txt, 1990.

[6] Steve Hilla. *The Extended Standard Product 3 Orbit Format (SP3-c).* http://igscb.jpl.nasa.gov/igscb/data/format/sp3c.txt, 2006.

[7] V.D. Scott and J. Clynch. A Proposed Standardized Exchange Format for Navstar GPS Geodetic Data. In *Proceedings of the Fourth International Geodetic Symposium on Satellite Systems*, Austin, Texas, April 1986.

Chapter 3

Converting Coordinates & Time

3.1 Transformations

Let $\mathbf{i}_x$, $\mathbf{i}_y$, $\mathbf{i}_z$ and $\mathbf{i}_\varepsilon$, $\mathbf{i}_\eta$, $\mathbf{i}_\zeta$ be two sets of orthogonal unit vectors

$$\mathbf{i}_\xi = l_1\mathbf{i}_x + m_1\mathbf{i}_y + n_1\mathbf{i}_z$$

$$\mathbf{i}_\eta = l_2\mathbf{i}_x + m_2\mathbf{i}_y + n_2\mathbf{i}_z$$

$$\mathbf{i}_\zeta = l_3\mathbf{i}_x + m_3\mathbf{i}_y + n_3\mathbf{i}_z$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} \varepsilon \\ \eta \\ \zeta \end{bmatrix} \text{ or } \begin{bmatrix} \varepsilon \\ \eta \\ \zeta \end{bmatrix} = \mathbf{R^T} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_x \cdot \mathbf{i}_\varepsilon & \mathbf{i}_x \cdot \mathbf{i}_\eta & \mathbf{i}_x \cdot \mathbf{i}_\zeta \\ \mathbf{i}_y \cdot \mathbf{i}_\varepsilon & \mathbf{i}_y \cdot \mathbf{i}_\eta & \mathbf{i}_y \cdot \mathbf{i}_\zeta \\ \mathbf{i}_z \cdot \mathbf{i}_\varepsilon & \mathbf{i}_z \cdot \mathbf{i}_\eta & \mathbf{i}_z \cdot \mathbf{i}_\zeta \end{bmatrix} = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix}$$

$$\mathbf{R^T} = \mathbf{R^{-1}}$$

Equations found here [1, pp. 81-82]

3.2 Time Systems

3.2.1 Solar & Sidereal Time

Since the beginning time has been kept by counting the days. An apparent solar day is the minimum time elapsed between the sun crossing a specified meridian and then recrossing the same meridian. This form of time keeping is problematic because no two apparent solar days are of the same duration due to Earth's rotation around the sun as well as around its axis (the Earth does a little more than one rotation per apparent solar day). Also, Earth's rotational speed is not constant and its axis of rotation is tilted 23.5° to the orbital plane. These imperfections call for correction, and thus mean solar time was created. A day in mean solar time is defined as one revolution of a hypothetical sun that orbits at the equator, and is more commonly known as Greenwich Mean Time. Another solution is to base our day on the crossing of a star much farther away thus minimizing the effect of the Earth's orbital movement, this method of time keeping is known as sidereal time. A sidereal day is about 4 minutes shorter than a solar day, and is used heavily by astronomers. Sidereal time is not truly stable either so mean sidereal day was introduced, and is known as Greenwich Apparent Sidereal Time. Universal Time (UT) refers to any time scale based on the Earth's rotation. UT0 refers to the mean solar time at the prime meridian as obtained from astronomical observation, and UT1 is UT0 corrected for polar motion. Briefly ephemeris time was introduced to standardize the second, which was defined as 1/31556925.9747 of the year 1900. This was soon replaced by atomic time [4, pp. 84-86].

3.2.2 Atomic Time

The second is now defined by an atomic standard that is based on the resonance frequency of the cesium atom. To be precise, the second is defined as "9,192,631,770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium-133 atom," whose duration happens to exactly match the ephemeris second discussed in the previous section. The problem with detaching our time keeping method from the Earth is that as the Earth slows its rotation noon will move closer to midnight (over the duration of thousands of years, of course). Coordinated Universal Time (UTC) was introduced to prevent this. UTC is a compromise between the precision of atomic time and the groundedness of Earth based time keeping, it uses the atomic second but introduces leap seconds (positive or negative) when necessary to keep UTC within .9 seconds of UT1 [4, pp. 86-87].

3.2.3 Time Formats

We are used to dealing with months, days, years, hours, minutes, and seconds, but such a time format makes for difficult epoch calculations over long periods. To solve this problem Julian Date (JD) was introduced. JD consists of a day count (days since noon UT on January 1^{st} 4713 B.C.) and a fraction of the current day. This makes for easy time differencing, but the length of the date can become cumbersome and the fact that a new day starts at noon confusing. To make things even easier Modified Julian Date (MJD) was created whose origin is midnight November 17^{th}, 1858.

$$\text{MJD} = \text{JD} - 2400000.5$$

In order to make Julian Date useful we need an easy way to go between calendar dates and JD. *timeconvert* does this and more with ease. The equations to convert from calendar date to JD are

$$\text{JD} = \text{INT}[365.25y] + \text{INT}[30.6001(m+1)] + D + \text{UT}/24 + 1720981.5$$

$$\begin{array}{lll} y = Y - 1 & \text{and } m = M + 12 & \text{if } M \leq 2 \\ y = Y & \text{and } m = M & \text{if } M > 2 \end{array}$$

where M is the month, D is the day, Y is the year, and INT[x] returns just the integer part of the number. To go from JD to calendar date

$$a = \mathrm{INT}[\mathrm{JD} + 0.5]$$

$$b = a + 1537$$

$$c = \mathrm{INT}[(b - 122.1)/365.25]$$

$$d = \mathrm{INT}[365.25c]$$

$$e = \mathrm{INT}[(b - d)/30.6001]$$

$$D = b - d - \mathrm{INT}[30.6001e] + \mathrm{FRAC}[\mathrm{JD} + 0.5]$$

$$M = e - 1 - 12\mathrm{INT}[e/14]$$

$$Y = c - 4715 - \mathrm{INT}[(7 + M)/10]$$

where FRAC[x] returns just the fractional part of a real number. MJD Conversion found here [4, p. 88]. All other date conversions were found here [2, pp. 36-37]

3.2.4 GPS Time

GPS Time (GPST) is a continuously running composite time kept by cesium and rubidium frequency standards aboard the satellites and at monitor stations. While there are no leap seconds in GPST as there are in UTC, it is steered to stay within 1 μs of UTC, that is the difference between GPST and UTC is an integer number of seconds plus a fraction of a μs. GPST is formatted in terms of GPS weeks and the number of seconds into the current week. Finding these values is done easily if the Julian Date is known.

$$\mathrm{GPS\ WEEK} = \mathrm{INT}[(\mathrm{JD} - 2444244.5)/7]$$

$$\mathrm{SOW} = \mathrm{FRAC}[(\mathrm{JD} - 2444244.5)/7] \times 604800$$

where INT[x] returns the integer part of a real number, FRAC[x] returns the fractional part, and SOW stands for Second of Week.

Other useful quantities such as Day of Week and Second of Day can be found using *time-convert* or the following equations.

$$\mathrm{DOW} = \mathrm{modulo}\{\mathrm{INT}[\mathrm{JD} + 0.5], 7\}$$

$$\mathrm{SOD} = \mathrm{modulo}\{\mathrm{FRAC}[\mathrm{JD} + 0.5], 7\} \times 86400$$

where DOW=0 corresponds to Monday, DOW=1 corresponds to Tuesday, and so on.

JD and GPS Week equations were found here [2, pp. 36-37], SOD derived from DOW equation.

3.2.5 Z-Count

Satellites keep internal time with Z-count, whose epoch period is 1.5 seconds (a convenient unit for communications timing). The full Z-count is 29 bits, the 10 bit GPS week folloed by a 19 bit Time of Week (TOW) expressed in Z-counts (or 1.5 second units). The truncated Z-count has a 17 bit TOW that is expressed in units of 6 seconds, or the length of one subframe's transmission time. Simply multiply the truncated TOW by 4 to get the full TOW [5, pp. 86-88].

$$\text{TOW} = \text{FRAC}[(\text{JD} - 2444244.5)/7] \times 403200$$

$$\text{Truncated TOW} = \text{FRAC}[(\text{JD} - 2444244.5)/7] \times 100800$$

Equations derived from SOW equation above

3.3 Earth Fixed Coordinates

3.3.1 ECI to ECF

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECF} = T^{xyz}_{XYZ} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{ECI}$$

$$T^{xyz}_{XYZ} = WSNP$$

P - applies precession, from epoch 2000.0 to the current time; N - applies nutation, from epoch 2000.0 to the current time; S - applies rotation to account for true sidereal time; W - applies polar motion;

Equations found on page 85 of Fundamentals of Orbit Determination paper book.

3.3.2 WGS-84

The World Geodetic System 1984 (WGS-84) is a fixed physical model of Earth produced by the Department of Defense to which many different reference frames can be attached. WGS-84 consists of two parts, a model of Earth's gravitational field, and an ellipsoid describing the Earth's general shape. When dealing with locations on the Earth's surface the ellipsoid provides the foundation for the geodetic coordinate system used by GPS. The ellipsoid's cross-sections parallel to the equatorial plane are circular while those orthogonal are elliptical. The ellipses are parameterized by an eccentricity e, a flattening f, and sometimes a second eccentricity e'

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

$$f = 1 - \frac{b}{a}$$

$$e' = \sqrt{\frac{a^2}{b^2} - 1} = \frac{a}{b}e$$

where a, the semimajor axis, is the value of the mean equatorial radius of Earth (6,378.137 km) and b, the semiminor axis, is the value of the polar radius of Earth (6,356.7523142 km) [3, pp. 25-26].

3.3.3 Coordinate Systems

Now that WGS-84 is defined it is important to understand what coordinate systems can be attached to the ellipsoid and how to move between these different systems. The GPS Toolkit comes with *poscvt*, an application that gives users the ability to easily convert coordinates in one reference frame to another. The coordinate systems that *poscvt* recognizes are Cartesian (or XYZ), geodetic, geocentric, and spherical coordinates. These systems and the formulas to convert between them are discussed below.

Cartesian (XYZ) Coordinates

The Earth Centered Earth Fixed (ECEF) Cartesian coordinate system is fixed to the WGS-84 ellipsoid and is the common ground that makes going between the Earth Centered Inertial (ECI) reference frame used by the satellites and the systems we are used to (such as latitude, longitude, and height) manageable. The equatorial plane makes the xy-plane with the $+x$-axis pointing toward $0°$ longitude and the $+y$-axis pointing toward $90°$ E longitude. The z-axis is normal to the equatorial plane and points to the geographical north pole. The conversion formulas presented in the next sections will convert to and from this Cartesian reference frame, and so to convert between two non-Cartesian coordinate systems the XYZ system will be used as an intermediary [3, p. 24].

Geodetic Coordinates

The geodetic coordinate parameters are longitude λ, latitude ϕ, and height h. Longitude is defined as the angle between the position and the x-axis in the equatorial plane, and is easily computed given a position in Cartesian coordinates. Let a user's position $\mathbf{U} = (x_u, y_u, z_u)$, then

$$\lambda = \begin{cases} \arctan\left(\frac{y_u}{x_u}\right), & x_u \geq 0 \\ 180° + \arctan\left(\frac{y_u}{x_u}\right), & x_u < 0 \text{ and } y_u \geq 0 \\ -180° + \arctan\left(\frac{y_u}{x_u}\right), & x_u < 0 \text{ and } y_u < 0 \end{cases}$$

where negative angles signal west longitude.

Latitude and height are not so straight forward. Latitude is determined by drawing a vector normal to the ellipsoid, beginning somewhere on the equatorial plane and terminating at the users position, we will call this the user vector. The smallest angle between this vector and the equatorial plane is the user's latitude, it is a North latitude for positive angles and South for negative. Notice that unless the user is at a pole or on the equator the vector does not pass through the center of the Earth. The users height is found by taking the magnitude of the vector originating on and normal to the ellipsoid and terminating at the user's position. Latitude ϕ and height h are found using the following equations

$$\phi = \arctan\left(\frac{z_u + e'^2 z_0}{r}\right)$$

$$h = U\left(1 - \frac{b^2}{aV}\right)$$

where

$$r = \sqrt{x_u^2 + y_u^2}$$

$$E^2 = a^2 - b^2$$

$$F = 54b^2 z_u^2$$

$$G = r^2 + (1 - e^2)z_u^2 - e^2 E^2$$

$$c = \frac{e^4 F r^2}{G^3}$$

$$s = \sqrt[3]{1 + c + \sqrt{c^2 + 2c}}$$

$$P = \frac{F}{3\left(s + \frac{1}{s} + 1\right)^2 G^2}$$

$$Q = \sqrt{1 + 2e^4 P}$$

$$r_0 = -\frac{Pe^2 r}{1 + Q} + \sqrt{\frac{1}{2}a^2\left(1 + \frac{1}{Q}\right) - \frac{P(1 - e^2)z_u^2}{Q(1 + Q)} - \frac{1}{2}Pr^2}$$

$$U = \sqrt{(r - e^2 r_0)^2 + z_u^2}$$

$$V = \sqrt{(r - e^2 r_0)^2 + (1 - e^2)z_u^2}$$

$$z_0 = \frac{b^2 z_u}{aV}$$

Going back to Cartesian coordinates from the geodetic system ($\lambda\ \phi\ h$) can be done more compactly

$$\mathbf{u} = \begin{bmatrix} \frac{a\cos\lambda}{\sqrt{1+(1-e^2)\tan^2\phi}} + h\cos\lambda\cos\phi \\ \frac{a\sin\lambda}{\sqrt{1+(1-e^2)\tan^2\phi}} + h\sin\lambda\cos\phi \\ \frac{a(1-e^2)\sin\phi}{\sqrt{1-e^2\sin^2\phi}} + h\sin\phi \end{bmatrix}$$

where $\mathbf{u}$ is the user's position vector [3, 4, pp. 26-28, p. 76].

Geocentric Coordinates

$$x = r\cos\phi\cos\lambda$$

$$y = r\cos\phi\sin\lambda$$

$$z = r\sin\phi$$

where λ and ϕ are geocentric longitude and latitude
found on page 82 in the Fundamentals of Orbital Determination paper book

Topocentric Coordinates

$$\mathbf{r}_t = T_t(\mathbf{r} - \mathbf{r}_s) = T_t\rho$$

$\mathbf{r}$ and $\mathbf{r}_s$ are the position vectors of the observer and satellite respectively in the Earth-fixed system

$$T_t = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ \cos\phi\cos\lambda & \cos\phi\sin\lambda & \sin\phi \end{bmatrix}$$

where λ and ϕ are geocentric longitude and latitude
found on page 84 in the Fundamentals of Orbital Determination paper book
to find *azimuth* (Az) and *elivation* (El)

$$\begin{array}{ll} \sin \mathrm{El} = \frac{z_t}{r_t} & -90^\circ \leq \mathrm{El} \leq 90^\circ \\ \sin \mathrm{Az} = \frac{x_t}{r_{xy}} & \\ \cos \mathrm{Az} = \frac{y_t}{r_{xy}} & 0^\circ \leq \mathrm{Az} \leq 360^\circ \end{array}$$

Equations found on pages 84-85 in Fundamentals of Orbit Determination paper book

3.4 References

[1] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics.* AIAA Press, Reston, Virginia, revised edition, 1999.

[2] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *GPS: Theory and Practice.* Springer-Verlag Wien, New York, NY, 5th edition, 2001.

[3] Elliot D. Kaplan, editor. *Understanding GPS: Principles and Applications.* Artech House Publishers, 685 Canton Street, Norwood, MA, 1996.

[4] Pratap Misra and Per Enge. *Global Positioning Sytem: Signals, Measurements and Performance.* Ganga-Jamuna Press, Lincoln, Massachusetts, 2004.

[5] James Bao-Yen Tsui. *Fundamentals of Global Positioning System Receivers: A Software Approach.* John Wiley & Sons, New York, 2000.

Part II

Usage, Examples & Notes

	Tool	Description	Execution Example
Transforms	calgps	generates a GPS calendar	`calgps -Y 2004`
	poscvt	converts a given input position to other position formats	`poscvt --geodetic="30.28 262.26700 167.64"`
	timeconvert	converts given input time to other time formats	`timeconvert --calendar="07 04 2006"`
	wheresat	outputs expected location of a satellite	`wheresat -b arl2100.06n -p 3`
Collecting & Converting	rtAshtech	records observations from an Ashtech receiver	`rtAshtech -p /dev/ttyS1 -o "minute%03j%02H%02m.%06yo"`
	ficfica ficafic fic2rin	convert fic files between ASCII, binary, and RINEX formats	`fic2rin fic2100.06 rin121.06n`
	mdp2fic mdp2rinex	convert MDP files to FIC or RINEX files	`mdp2rinex -i mdpfile -o arl2100.06o`
	novaRinex	convert Novatel files to RINEX	`novaRinex --input nova2100.06 --obstype L1`
	navdmp	dumps information from nav files to human readable formats	`navdmp -i arl2100.06n -o arl2100.06.dmp`
	RinexDump	dumps observation data for specified satellites from a RINEX file	`RinexDump arl2100.06o 3 4 L1 L2`
Comparing & Validating	ephdiff	compares the satellite positions from two ephemeris sources	`ephdiff arl2100.06n fic2100.06`
	ficdiff	compares contents of two FIC files	`ficidff fic12100.06 fic22100.06`
	ficcheck ficacheck	reads a FIC file and checks it for errors reporting the first found	`ficcheck fic2100.06 -t "07/20/2006 11:00:00"`
	rowdiff rnwdiff rmwdiff	compares contents of two RINEX files	`rowdiff arl1210.06o arl22100.06o`
	rowcheck rnwcheck rmwcheck	reads RINEX files and checks for errors reporting the first found	`rnwcheck arl210.06n -e "07/20/2006 11:00:00"`
	navsum RinSum	summarizes the contents of nav/RINEX files	`RinSum -i arl2100.06o --EpochBeg 2006,07,20,13,20,00`
	mdptool	summarizes MDP data	`mdptool -i mdpfile --pvt --obs`
Reszilla	ddGen	computes double-difference residuals from raw observations	`ddGen -1 arl2100.09o -2 arl2110.09o -e arl2100.09n`
	ordClock	generates clock estimates for each epoch of ords	`ordClock -i ord.out -t "%4Y %3j"`
	ordEdit	edits an ord file based on various criteria	`ordEdit -i ord.out -c -s 0.5 -t "%4Y %3j"`
	ordLinEst	computes a linear clock estimate	`ordLinEst -i ord.out -t "%4Y %3j" --ns`
	ordStats	computes ords statistics	`ordStats -i ord.out -b 0-10`
	ordGen	generates observed range deviations	`ordGen -o arl2100.09o -e arl2100.09n -t "%04Y %03j"`

Table 3.1: GPSTk Applications, categorized, with execution examples.

	Tool	Description	Execution Example
RINEX Tools	RinDump	dumps observation data for specified satellites from a RINEX file	`RinDump --obs arl2100.09o`
	RinSum	provides a summary of an input RINEX file	`RinSum --file arl2100.09o`
	RinNav	reads one or more Rinex Nav files and merges the navigation data to a single	`RinNav --file brdc0300.02n`
	RinEdit	opens, edits, and outputs a single RINEX file from one or more input	`RinEdit --IF ARL82660.09o --OF obsOut.04o`
Editing Data	mergeFIC	sorts and merges input FIC files into a single file	`mergeFIC -i fic12100.06 -i fic22100.06 -o ficmerge2100.06`
	mergeRinObs, -Nav, -Met	sorts and merges RINEX files	`mergeRinNav -i arl2100.06n -i arl2110.06n arl210-211.06n`
	NavMerge	merges RINEX nav files into a single file	`NavMerge -oarlnavs.06n arl2100.06n arl2110.06n`
	rinexthin	decimates an input RINEX observation files to desired data rate	`rinexthin -f arl2100.06o -s 30 -o arl2100thin.06n`
	ResCor	edits RINEX files and computes corrections	`ResCor -IFarl2100.06o -OFarl2100mod.06o -DS12,12:00:00`
	DiscFix	cycle slip corrector	`DiscFix --inputfile arl2100.06o --dt 1.5`
Iono	IonoBias	solves interfrequency biases and provides a simple ionosphere model	`IonoBias --input arl2100.06o --nav arl2100.06n --XSat 3`
	TECMaps	creates maps of Total Electron Content (TEC)	`TECMaps --input arl2100.06o --nav arl2100.06n --LinearFit`
Positioning	PRSolve	generates autonomous position solution	`PRSolve -o arl2100.06o -n arl2100.06nn --XPRN 12`
	rinexpvt	generates autonomous position solution	`rinexpvt -o alr2100.06o -n arl2100.06n`
	DDBase	computes a network solution using carrier phase	`DDBase ... --ObsFile arl2100.06o --PosXYZ x,y,z,1 --Fix`
	vecsol	estimates short baseline using range or carrier phase	`vecsol station12100.06o station22100.06o`

Table 3.1: GPSTk Applications, continued.

3.5 *ash2mdp ash2xyz*

3.5.1 Overview

These applications process Ashtech Z(Y)-12 observation and ephemeris data and output satellite positions and ionospheric corrections in either MDP or XYZ format.

3.5.2 Usage

ash2mdp ash2xyz

Optional Arguments

Short Arg.	Long Arg.	Description
-i		Where to get data from. The default is to use stdin.
-o		Where to send the output. The default is to use stdout.
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-w	–week=NUM	The full GPS week in which this data starts. Use this option when the start time of the data being processed is not during this week.
-c	–code=ARG	Restriction for source of observation data collected via L1/L2 Y code tracking will be used. Options are ”Y”, ”P”, and ”codeless.” XYZ only.
-s	–offset=NUM	Output SV positions at a time offset from the current time. Give a positive or negative integer of seconds. XYZ only.
-n	–num_points=NUM	Width of the exponential filter moving window, in number of points (default is 36). XYZ only.

3.5.3 Notes

Input is on the command line, or of the same format in a file (-f<file>).

3.6 *bc2sp3*

3.6.1 Overview

This application reads RINEX navigation file(s) and writes to SP3 (a or c) file(s).

3.6.2 Usage

bc2sp3

Optional Arguments

Short Arg.	Long Arg.	Description
	–in	Read the input file (repeatable).
	–out	Name the output file. Default is sp3.out.
	–tb	Output beginning epoch; <time> = week, sec-of-week (earliest in input).
	–te	Output ending epoch; <time> = week, sec-of-week (latest in input).
	–outputC	Output version c (no correlation) (otherwise a).
	-msg	Add message as a comment to the output header (repeatable).
	–verbose	Output to screen: dump headers, data, etc.
	–help	Print this message and quit.

3.6.3 Examples

```
 bc2sp3 --in nav/s121001a.00n --in nav/s121001a.01n --out bc2sp3.out --verbose
Reading file nav/s121001a.00n
Input--------------------------------- REQUIRED ----------------------------------
Rinex Version  2.10,  File type Navigation.
Prgm: RinexNavWriter,  Run: 11-08-01  0:31:01,  By: NIMA
(This header is VALID 2.11 Rinex.)
--------------------------------- OPTIONAL ---------------------------------
 Ion alpha is NOT valid
 Ion beta is NOT valid
 Delta UTC is NOT valid
 Leap seconds is NOT valid
-------------------------------- END OF HEADER -------------------------------
Reading file nav/s121001a.01n
Input--------------------------------- REQUIRED ----------------------------------
Rinex Version  2.10,  File type Navigation.
Prgm: RinexNavWriter,  Run: 11-08-01  0:31:02,  By: NIMA
(This header is VALID 2.11 Rinex.)
--------------------------------- OPTIONAL ---------------------------------
 Ion alpha is NOT valid
 Ion beta is NOT valid
 Delta UTC is NOT valid
 Leap seconds is NOT valid
-------------------------------- END OF HEADER -------------------------------
```

```
SP3 Header: version SP3a containing positions and velocities.
 Time tag : 2000/01/01  0:14:44
 Timespacing is 900.00 sec, and the number of epochs is 208
 Data used as input : BCE
 Coordinate system : WGS84
 Orbit estimate type :
 Agency : ARL
 List of satellite PRN/accuracy (30 total) :
 G01/0 G02/0 G03/0 G04/0 G05/0 G06/0 G07/0 G08/0
 G09/0 G10/0 G11/0 G13/0 G14/0 G15/0 G16/0 G17/0
 G18/0 G19/0 G20/0 G21/0 G22/0 G23/0 G24/0 G25/0
 G26/0 G27/0 G28/0 G29/0 G30/0 G31/0
 Comments:
End of SP3 header
* G01 2000/01/01  0:14:44.000 = 1042/519284.000
P G01 2000/01/01  0:14:44.000 = 1042/519284.000 X=  25704.923932
Y=   1917.715173 Z=  -6382.182137 C=      0.010948 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
V G01 2000/01/01  0:14:44.000 = 1042/519284.000 X=     73.647819
Y=     46.729037 Z=    302.940947 C=      0.000000 sX= 0 sY= 0 sZ= 0 sC=  0
P G03 2000/01/01  0:14:44.000 = 1042/519284.000 X=  19615.286679
Y=  13022.977045 Z= -12340.096622 C=      0.001460 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
V G03 2000/01/01  0:14:44.000 = 1042/519284.000 X=   -158.845279
Y=     -3.592649 Z=   -256.800421 C=      0.000000 sX= 0 sY= 0 sZ= 0 sC=  0
P G14 2000/01/01  0:14:44.000 = 1042/519284.000 X=  21304.591776
Y=  -7854.561000 Z=  13783.692368 C=     -0.001147 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
V G14 2000/01/01  0:14:44.000 = 1042/519284.000 X=   -112.966658
Y=    134.498918 Z=    250.863009 C=      0.000000 sX= 0 sY= 0 sZ= 0 sC=  0
P G15 2000/01/01  0:14:44.000 = 1042/519284.000 X=  15085.444070
Y=  12582.798439 Z=  17649.742134 C=      0.010795 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
V G15 2000/01/01  0:14:44.000 = 1042/519284.000 X=     39.944949
Y=    225.075281 Z=   -191.841184 C=      0.000000 sX= 0 sY= 0 sZ= 0 sC=  0
P G16 2000/01/01  0:14:44.000 = 1042/519284.000 X=  19460.508602
Y= -17881.770281 Z=   1051.372781
0C=     -0.002944 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
...
```

3.7 *CalcDOPs*

3.7.1 Overview

This application reads SV almanac data (one file per day of observation) from a FIC, FICA or a RINEX navigation file, then computes and displays visibility information. Dilution of precision values from that data are calculated using standard methods. See for example:

- AIAA GPS Theory and Applications vol. 1, Ed. Parkinson & Spilker, pp. 414.
- GPS Signals, Measurements, and Performance, 2ed., Misra & Enge, pp. 203.

3.7.2 Usage

CalcDOPs

Required Arguments

Short Arg.	Long Arg.	Description
-i<inputfile>		Input file for day to be calculated.

Optional Arguments

-p <inputfile>		Input file for previous day (ephemeris mode only).
-o <outputfile>		Grid output file (default DOPs.out).
-sf <outputfile>		Stats output file (default DOPs.stat).
-tf <outputfile>		Time steps output file (default DOPS.times).
-l <outputfile>		Log output file (default DOPS.log).
-rs		Read from stats file.
-a		Work in almanac mode (ephemeris mode is default).
-w -s <week> <sow>		Starting time tag.
-x <prn>		Exclude satellite PRN.
-t <dt>		Time spacing.
-na		North America only.
-d		Dump grid results at each time step (time-intensive).
-h	–help	Output options info and exit.
-v		Print version info and exit.

3.7.3 Notes

Abort/failure codes given on return:

-1	could not open input data file
-2	could not identify input data file type
-3	fewer than 4 satellite almanacs available
-4	could not allocate GridStats data types
-5	could not open input stats file
-6	could not open output grid file
-7	could not open output stats file
-8	could not open output log file

Essential variables not documented below at declaration:

NtrofN	number of cells/times with < 5 SVs visible during the time period
NpeakH	number cells/times w/ HDOP > 10
NpeakP	number cells/times w/ PDOP > 10
IworstN	index in Grid[] of cell with worst nsvs (number of satellites)
IworstH	index in Grid[] of cell with worst HDOP
IworstP	index in Grid[] of cell with worst PDOP
WorstN	value of nsvs at IworstN
WorstH	value of HDOP at IworstH
WorstP	value of PDOP at IworstP
TworstN	time tag (CommonTime class) of WorstN
TworstH	time tag (CommonTime class) of WorstH
TworstP	time tag (CommonTime class) of WorstP

1. GPS only, using PRNs hard-wired to SV numbers 1-32.
2. Elevation limit is hard-wired to 5 degrees above horizon.
3. "North America" means the northern half-hemisphere: -180 to 0 deg long., 0 to 90N latitude.
4. Ephemeris mode is default, almanac mode is optional. Ephemeris mode is preferred, because it excludes unhealthy satellites for any time when they transmitted an unhealthy flag. Almanac mode will generally not exclude SVs when they were unhealthy (typical), or may erroneously exclude them for an entire day (rarely).
5. If 2 input files are given, the default start time is midnight on the day to be calculated. A previous-day input file can be given only in ephemeris mode, not almanac.
6. The code uses geodetic coordinates for all calculations.
7. The -d option is useful for (e.g.) making movies of DOPs throughout a day.

3.7.4 Examples

```
> CalcDOPs -i nav/s121001.02n -d

-------------------------------DOPs.out file-------------------------------------

  -180.000 -89.000   5.574   4.978   2.701   3.950   2.443  5.878  80.406  76.501
73.222  47.296  40.713  4.000 0.0937500
  -120.000 -89.000   5.489   4.899   2.664   3.879   2.410  5.882  81.008  77.069
73.745  47.482  41.165  4.000 0.1006944
  -60.000 -89.000   5.338   4.767   2.619   3.752   2.342  5.910  80.702  76.846
73.858  47.931  41.151  4.000 0.0868056
    0.000 -89.000   5.197   4.637   2.569   3.628   2.285  5.951  79.798  76.057
73.423  48.182  40.680  4.000 0.0833333
   60.000 -89.000   4.788   4.259   2.430   3.280   2.132  5.965  79.208  75.499
72.903  48.004  40.246  4.000 0.0763889
```

```
  120.000 -89.000   4.814   4.284   2.433   3.322   2.139  5.948  79.510  75.720
72.814  47.567  40.266  4.000 0.0798611
```

3.8 *calgps*

3.8.1 Overview

This application generates a dual GPS and Julian calendar to either stdout or to a graphics file. The arguments and format are inspired by the UNIX 'cal' utility. With no arguments, the current argument is printed. The last and next month can also be printed. Also, the current or any given year can be printed.

3.8.2 Usage

calgps

Optional Arguments

Short Arg.	Long Arg.	Description
-h	–help	Generates help output.
-3	–three-months	Prints a GPS calendar for the previous, current, and next month.
-y	–year	Prints a GPS calendar for the entire current year.
-Y	–specific-year=NUM	Prints a GPS calendar for the entire specified year.
-p	–postscript=ARG	Generates a postscript file.
-s	–svg=ARG	Generates an SVG file.
-e	–eps=ARG	Generates an encapsulated postscript file.
-v	–view	Try to launch an appropriate viewer for the file.
-n	–no-blurb	Suppress GPSTk reference in graphic output.

3.8.3 Examples

```
> calgps -3

                       Jun 2011
1638                            1-152  2-153  3-154  4-155
1639   5-156  6-157  7-158  8-159  9-160 10-161 11-162
1640  12-163 13-164 14-165 15-166 16-167 17-168 18-169
1641  19-170 20-171 21-172 22-173 23-174 24-175 25-176
1642  26-177 27-178 28-179 29-180 30-181

                       Jul 2011
1642                                          1-182  2-183
1643   3-184  4-185  5-186  6-187  7-188  8-189  9-190
1644  10-191 11-192 12-193 13-194 14-195 15-196 16-197
1645  17-198 18-199 19-200 20-201 21-202 22-203 23-204
1646  24-205 25-206 26-207 27-208 28-209 29-210 30-211
1647  31-212

. . .
```

3.8.4 Notes

If multiple options are given only the first is considered.

3.9 *compSatVis compStaVis*

3.9.1 Overview

compSatVis computes satellite visibility. *compStaVis* computes station visibility.

3.9.2 Usage

compSatVis compStaVis

Required Arguments

Short Arg.	Long Arg.	Description
-o	–output-file=ARG	Name of the output file to write.
-n	–nav=ARG	Name of navigation file.
-c	–mscfile=ARG	Name of MS coordinates file.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-p	–int=ARG	Interval in seconds.
-e	–minelv=ARG	Minimum elevation angle.
-t	–navFileType=ARG	FALM (FIC Almanac), FEPH (FIC Ephemeris), RNAV, YUMA, SEM (System Effectiveness Model), or SP3.
-m	–min-sta=ARG	Minimum number of stations visible simultaneously. compStaVis only.
-m	–max-SV=ARG	Maximum number of SVs tracked simultaneously. compSatVis only.
-D	–detail	Print SV count for each interval.
-x	–exclude=ARG	Exclude station.
-i	–include=ARG	Include station.
-s	–start-time=TIME	Start time of evaluation (”m/d/y H:M”).
-z	–end-time=TIME	End time of evaluation (”m/d/y H:M”).

3.9.3 Examples

Generating satellite visibility statistics using the SEM almanac from the USCG Navigation Center.

This example loads SEM almanac data from the file current.al3 and a list of station locations from the file stations.msc. It then calculates the number of satellites visible to each station found at each 60 sec interval from 0000Z to 2356Z of Jan 13, 2008. using a 10 degree minimum elevation angle. The results are written to the file visout.txt. Note the use of a specific start time. The SEM and Yuma almanac formats contain an almanac reference week, which is generally in the range 0-1023 (the existing format definitions are ambiguous and SEM and Yuma almanacs with full week numbers have been reported, at least anecdotally). If the -s command is not specified, compSatVis will use whatever reference time is given in the almanac file, which may result in unexpected results.

```
user@host:~$ compSatVis -ovisout.txt -ncurrent.al3 -tSEM
```

```
-cstations.msc -e10 -p60 -s"01/16/2008 00:00"
```

Generating station visibility statistics using the SEM almanac from the USCG Navigation Center.

Same as the previous example, however, the values calculated and the statistics will reflect the number of stations visible to each satellite.

```
user@host:~$ compSatVis -ovisout.txt -ncurrent.alm -tYUMA
-cstations.msc -e10 -p60 -s"01/13/2008 00:00" -z"01/16/2008 23:59"
```

Generating satellite visibility statistics using the Yuma almanac from the USCG Navigation Center.

Similar to the first example, but the statistics are computed over four complete days.

```
user@host:~$ compSatVis -ovisout.txt -ncurrent.alm -tYUMA
-cstations.msc -e10 -p60 -s"01/13/2008 00:00" -z"01/16/2008 23:59"
```

Generating satellite visibility statistics using SP3 files.

Similar to the first example, however, navigation message data are from three SP3 files. It is necessary to load three SP3 files to cover the default sidereal day period because the methods that calculate SV positions from the SP3 data use interpolation and need data from the previous day and the following day in order to have sufficient points for the interpolation. In this example in which no evaluation period is specified, compSatVis derives coverage for the "middle day" for the period.

```
user@host:~$ compSatVis -ovisout.txt -napc14622 -napc14623 -napc14624
-tSP3 -cstations.msc -e10 -p60
```

3.10 *ConstellationList*

3.10.1 Overview

ConstellationList provides lists of the GPS SV PRN ID active/inactive on a given day.

3.10.2 Usage

ConstellationList

Required Arguments

Short Arg.	Long Arg.	Description
-i	–input-file=<arg>	The name of the Constellation Definition file(s) to read.
-y	–year=<arg>	Year of interest.
-j	–day-of-year=<arg>	Day of year.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-O	–OpsAd	Assume input file is Op Advisory format (CSV is default).
-b	–Base24	List PRNs in Base 24 Constellation.
-x	–excessSVs	List PRNs in use, but in excess of the Base 24 Constellation.
-n	–notBase24	List PRNs NOT used in Base 24 Constellation.
-s	–SVN Output	Output SVN in place of PRN (not valid for -O).

3.10.3 Examples

```
>ConstellationList -iSlot2008.csv -tC -y2008 -j001 -b
2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 27, 28, 30, 31
>ConstellationList -iSlot2008.csv -tC -y2008 -j001 -n
1, 5, 7, 24, 25, 26, 29, 32
>ConstellationList -iSlot2008.csv -tC -y2008 -j001 -n
1, 5, 24, 25, 26
>ConstellationList -iSlot2008.csv -tC -y2008 -j001 -b -s
61, 33, 34, 36, 38, 39, 40, 46, 58, 43, 41, 55, 56, 53, 54, 59, 51, 45, 47, 60, 27, 44, 30, 52
```

3.10.4 Notes

In particular, ConstellationList provides a means of determining which SVs are members of the “Base 24” constellation and which are not. This is usually of no interest to the general user, but is important in cases where programs are evaluating GPS performance against a defined standard such at the Standard Positioning Service (SPS) Performance Standard (PS) which defines GPS performance in terms of the ”official” constellation as opposed to the superset which is normally available. The results are provided as a text list of comma separated values on a a single line, suitable for piping into another process.

The complication in this process is that the information regarding the orbit plane/slot of each SV is not available from the broadcast message. It must be obtained “external to the system". One source of such information is the USCG Navigation Center website which stores the Operational Advisories. These advisories provide the relationship between SVs and plane/slot assignments. ConstellationList is programmed to read the advisories as an input format, as long as the format of the advisories does not change.

As an alternative to the Operational Advisories, ARL:UT has prepared files of the assignments for specific years as comma separated value files. Each line in these files represents the status on a given day and includes the mapping between the PRN IDs and the NAVSTAR numbers. These files have been hand-checked and are available in the GPSTk repository as Slot2007.csv and Slot2008.csv.

3.11 *daa*

3.11.1 Overview

This application performs a data availability analysis of the input data. In general, availability is determined by station and satellite position.

3.11.2 Usage

daa

Required Arguments

Short Arg.	Long Arg.	Description
-e	–eph=ARG	Where to get the ephemeris data. Acceptable formats include RINEX nav, FIC, MDP, SP3, YUMA, and SEM. Repeat for multiple files.
-o	–obs=ARG	Where to get the observation data. Acceptable formats include RINEX obs, MDP, smooth, Novatel, and raw Ashtech. Repeat for multiple files. If a RINEX obs file is provided, the position will be taken from the header unless otherwise specified.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
	–ouput=ARG	Output location (default is stdout).
-x	–independent=ARG	The independent variable in the analysis. The default is time.
-c	–msc=ARG	Station coordinates file.
-m	–msid=ARG	Station for which to process data. Used to select a station position from the msc file.
-t	–time-format=ARG	CommonTime format specifier used for times in the output. The default is "%Y %j %02H:%02M:%04.1f".
	–mask-angle=ARG	Ignore anomalies on SVs below this elevation. The default is 10 degrees.
	–track-angle=ARG	Assume the receiver starts tracking at this elevation. The default is 10 degrees.
	–time-mask=ARG	Ignore anomalies on SVs that haven't been above the mask angle for this number of seconds. The default is 0 seconds.
	–snr=ARG	Discard data with an SNR less than this value. The default is 20 dB-Hz.
-p	–position=ARG	Receiver antenna position in position (x,y,z) coordinates. Format as a string: "X Y Z".
-l	–time-span=ARG	How much data to process, in seconds.
	–ignore-prn=ARG	Specify the PRN of an SV to not report on in the output. Repeat to specify multiple SVs.
	–obs-interval=ARG	Specify the time interval, in seconds, between observations. The default is to scan the file to discover this via examination of the file.
-b	–bad-health	Ignore anomalies associated with SVs that are marked unhealthy.
-s	–smash-adjacent	Combine adjacent lines from the same PRN.

–start-time=TIME	Ignore data before this time.
–stop-time=TIME	Ignore any data after this time.

3.11.3 Examples

```
> daa -o s121001a.05o -e s121001a.05n

 Availability Raw Results :

Start                End         #     PRN    Elv          Az  Hlth  ama ata
============================================================================
2005 1 00:06:30.0            1     4    10.03^        316   0     8   9
2005 1 00:07:00.0            1     4    10.20^        316   0     8   9
2005 1 00:07:30.0            1     4    10.38^        316   0     8   9
2005 1 00:08:00.0            1     4    10.55^        315   0     8   9
2005 1 00:08:30.0            1     4    10.73^        315   0     8   9
2005 1 00:09:00.0            1     4    10.91^        315   0     8   9
2005 1 00:12:30.0            1     4    12.13^        314   0     8   9
+L1 GPSP,L2 GPSP
2005 1 00:26:30.0            1     24   10.36^        313   0     8   9
+L1 GPSP,L2 GPSP
2005 1 02:02:30.0            1     11   10.02v        123   0     7   9
2005 1 03:09:00.0            1     20   10.11v        141   0     8   10
...
2005 1 22:28:00.0            1     20   10.50^         24   0     7   9
2005 1 22:30:00.0            1     20   11.04^         25   0     7   9
2005 1 22:30:30.0            1     20   11.17^         25   0     7   9
2005 1 22:31:00.0            1     20   11.31^         25   0     7   9
2005 1 22:33:00.0            1     20   11.86^         26   0     7   9
+L1 GPSP,L2 GPSP
2005 1 22:49:30.0            1     3    10.00v        117   0     7   9
2005 1 22:52:00.0            1     7    10.04^        273   0     7   9
+L1 GPSP,L2 GPSP

 Summary:

Analysis span: 2453372 00000000 0.000000000000000 GPS through 2453372 86370000
0.000000000000000 GPS
Data span:     2453372 00000000 0.000000000000000 GPS through 2453372 86370000
 0.000000000000000 GPS
Total number of epochs with data: 2880
Epochs with any data missing: 184
Epochs without data from any SV: 0
SV-Epochs expected: 23234
Channel Loss: 0.00000 % (0)
SV-Epochs missed: 0.71017 % (165)
```

3.12 *DiscFix*

3.12.1 Overview

This application reads a RINEX observation data file containing GPS dual-frequency pseudorange and carrier phase measurements, divides the data into 'satellite passes', and finds and fixes discontinuities in the phases for each pass.

Output is a list of editing commands for use with program EditRinex. DiscFix will (optionally) write the corrected pseudorange and phase data to a new RINEX observation file. Other options will also smooth the pseudorange and/or debias the corrected phase.

DiscFix calls the GPSTk Discontinuity Corrector (GDC vers 5.3 7/14/2008).

3.12.2 Usage

DiscFix

Required Arguments

Short Arg.	Long Arg.	Description
	–inputdir	File containing more options.
	–dt	Time space in seconds of the data.

Optional Arguments

Short Arg.	Long Arg.	Description
-f	–file	File containing more options.
	–beginTime	Start time of processing (BOF).
	–endTime	End time of processing (EOF).
	–decimate	Decimate data to specified time interval, in seconds.
	–forceCA	Use C/A code range, NOT P code. Default only if P absent.
	–gap	Minimum data gap in seconds separating satellite passes (600).
	–onlySat	Process only satellite (GPS SatID, e.g. G21).
	–exSat	Exclude satellite(s) (GPSSatID).
	–smoothPR	Smooth pseudorange and output in place of raw pseudorange.
	–smoothPH	Debias phase and output in place of raw phase.
	–smooth	Same as –smoothPR AND –smoothPH.
	–DClabel	Set Discontinuity Corrector parameter 'label' to 'value'.
	–DChelp	Print a list of GDC parameters and their defaults, then quit.
	–logOut	Output log file name (df.log).
	–cmdOut	Output file name, for editing commands (df.out).
	–format	Output time format (gpstk::CommonTime) (%4F %10.3g).
	–RinexFile	RINEX (obs) file name for output of corrected data.
	–RunBy	RINEX header 'RUN BY' string for output.
	–Observer	RINEX header 'OBSERVER' string for output.
	–Agency	RINEX header 'AGENCY' string for output.
	–Marker	RINEX header 'MARKER' string for output.
	–Number	RINEX header 'NUMBER' string for output.
-h	–help	Print this syntax page and quit.

–verbose	Print extended output to the log file.

3.12.3 Examples

```
> DiscFix --dt 1.5 --inputfile ar12800.06o

DiscFix, part of the GPS ToolKit, Ver 5.0 8/20/07, Run 2011/07/22 11:17:25
DiscFix is writing to log file df.log
DiscFix is writing to output file df.out
DiscFix timing: 0.960 seconds.
```

3.13 *DOPcalc*

3.13.1 Overview

This application computes position, time, and geometric dilution of precision (DOP) parameters.

3.13.2 Usage

DOPcalc

Required Arguments

Short Arg.	Long Arg.	Description
-e	–eph=ARG	Where to get the ephemeris data. Acceptable formats include RINEX nav, FIC, MDP, SP3, YUMA, and SEM. Repeat for multiple files.
-o	–obs=ARG	Where to get the observation data. Acceptable formats include RINEX obs, MDP, smooth, Novatel, and raw Ashtech. Repeat for multiple files. If a RINEX obs file is provided, the position will be taken from the header unless otherwise specified.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-p	–position=ARG	User position in ECEF (x,y,z) coordinates. Format as a string: "X Y Z".
	–el-mask=ARG	Elevation mask to apply, in degrees. The default is 0.
-c	–msc=ARG	Station coordinate file.
-m	–msid=ARG	Monitor station ID number.

3.13.3 Examples

```
> DOPcalc -o s121001a.05o -e s121001a.05n

>Time    # SVs    GDOP     PDOP     TDOP
2005/001/00:00:00.0   7 20618758.65 20618758.65     0.00
2005/001/00:00:30.0   7     3.58     3.09     1.34
2005/001/00:01:00.0   7     3.58     3.09     1.34
2005/001/00:01:30.0   7     3.58     3.09     1.34
2005/001/00:02:00.0   8     2.54     2.26     1.08
2005/001/00:02:30.0   8     2.56     2.27     1.08
...
```

3.14 *ephdiff*

3.14.1 Overview

The application compares the contents of two files containing ephemeris data.

3.14.2 Usage

ephdiff

%multicolumn3c*ephdiff*

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-f	–fic=ARG	Name of an input FIC file.
-r	–rinex=ARG	Name of an input RINEX NAV file.

3.14.3 Examples

```
> ephdiff -f fic06.187 -r arl2800.06n

Broadcast Ephemeris (Engineering Units)

PRN : 11

              Week(10bt)     SOW     DOW   UTD     SOD   MM/DD/YYYY   HH:MM:SS
Clock Epoch:  1382( 358)  417600   Thu-4   187   72000   07/06/2006   20:00:00
Eph Epoch:    1382( 358)  417600   Thu-4   187   72000   07/06/2006   20:00:00
Transmit Week:1382
Fit interval flag :  0

          SUBFRAME OVERHEAD

               SOW    DOW:HH:MM:SS     IOD    ALERT   A-S
SF1 HOW:    411426  Thu-4:18:17:06    0x17D      0     on
SF2 HOW:    411432  Thu-4:18:17:12     0x7D      0     on
SF3 HOW:    411438  Thu-4:18:17:18     0x7D      0     on

           CLOCK
. . .
```

3.14.4 Notes

Both files can either be a RINEX or a FIC file.

3.15 *ephsum*

3.15.1 Overview

ephsum summarizes contents of a RINEX navigation message or FIC file and outputs to text file. The summary contains the transmit time, time of effectivity, end of effectivity, IODC, and health as a one-line-per ephemeris summary. The number of ephemerides found per SV is also provided. The number of ephemerides per SV is also summarized at the end. The default is to summarize all SVs found. If a specific PRN ID is provided, only data for that PRN ID will be summarized.

3.15.2 Usage

ephsum

Required Arguments

Short Arg.	Long Arg.	Description
-i	–input-file=ARG	Input file name(s).
-o	–output-file=ARG	Output file name.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-p	–PRNID=ARG	The PRN ID of the SV to process (default is all SVs).
-x	–xmit	List in order of transmission (default is TOE).

3.15.3 Examples

```
> ephsum -i nav/s121001a.01n -o ephsumOut.txt

# Output file from EphSum
# Processing input specification: nav/s121001a.01n - Success(RINEX)
#
#PRN: 01,  # of eph: 08
#PRN !               Xmit                !             Toe/Toc
!            End of Eff             !  IODC   Health
  01 ! 1095  86400 01/01/01 001 00:00:00 ! 1095  93600 01/01/01 001 02:00:00
! 1095 100800 01/01/01 001 04:00:00 ! 0x088  0x00 00
  01 ! 1095  93600 01/01/01 001 02:00:00 ! 1095 100800 01/01/01 001 04:00:00
! 1095 108000 01/01/01 001 06:00:00 ! 0x089  0x00 00
  01 ! 1095 107940 01/01/01 001 05:59:00 ! 1095 108000 01/01/01 001 06:00:00
! 1095 122340 01/01/01 001 09:59:00 ! 0x28A  0x00 00
  01 ! 1095 108000 01/01/01 001 06:00:00 ! 1095 115200 01/01/01 001 08:00:00
! 1095 122400 01/01/01 001 10:00:00 ! 0x18B  0x00 00
  01 ! 1095 115200 01/01/01 001 08:00:00 ! 1095 122400 01/01/01 001 10:00:00
! 1095 129600 01/01/01 001 12:00:00 ! 0x186  0x00 00
  01 ! 1095 160500 01/01/01 001 20:35:00 ! 1095 165600 01/01/01 001 22:00:00
! 1095 174900 01/02/01 002 00:35:00 ! 0x286  0x00 00

...
```

```
#
#PRN: 32,  # of eph: NONE
#
#Summary of Counts by PRN
# PRN    Count
#  01       8
#  02       5
#  03       5
#  04       6
#  05       8
#  06       7
#  07       5
#  08       7
#  09       6
#  10       5
#  11       5
#  12       0
#  13       8
#  14       5
#  15       7
#  16       0
#  17       5
#  18       0
```

3.16 *fic2rin*

3.16.1 Overview

This application converts navigation messages between the FIC format, a format for GPS observations established by ARL:UT, and the RINEX format.

3.16.2 Usage

fic2rin

```
fic2rin usage: fic2rin <input FIC file> <output RINEX file name>
```

3.16.3 Examples

```
> fic2rin fic06.187 rin1870.06

sh: fic2rin: not found
```

```
File Snippets

Binary FIC File

0000000
*
0000020                                    B   L   K       m  \0  \0  \0
0000030  \0  \0  \0  \0      \0  \0  \0  \0  \0  \0  \0   f 005  \0  \0
0000040 022  \0  \0  \0   >   f 301   " 260   i   {   !   f  \0   d 026
0000050 335 344   8  \t 002   b   C 035 205   7   4 027 241 372 210 006
0000060 006   }   Y   / 301 374   ?  \0   \   S 021   8   >   f 301   "
 . . .

RINEX NAV File

     2.10           NAVIGATION                              RINEX VERSION / TYPE
fic2rin                         07/13/2006 11:48:58 PGM / RUN BY / DATE
                                                            END OF HEADER
 5 06  7  6 19 59 44.0  .199091155082D-03  .356976670446D-10  .000000000000D+00
     .118000000000D+03 -.656250000000D+00  .538879589355D-08  .997594152841D+00
    -.409781932831D-07  .710751442239D-02  .655464828014D-05  .515355578804D+04
     .417584000000D+06 -.104308128357D-06 -.249936238139D+01  .707805156708D-07
     .938194464982D+00  .241750000000D+03  .105751234129D+01 -.843570852398D-08
     .600024993449D-10  .100000000000D+01  .138200000000D+04  .000000000000D+00
     .240000000000D+01  .000000000000D+00 -.419095158577D-08  .118000000000D+03
     .411426000000D+06  .400000000000D+01
 . . .
```

3.17 *ficacheck ficcheck*

3.17.1 Overview

These applications read input ASCII or binary FIC and check them for errors. *ficcheck* checks binary files and *ficacheck* checks ASCII files.

3.17.2 Usage

ficacheck ficcheck

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-t	–time=TIME	Time of first record to count (default BOT).
-e	–end-time=TIME	End of time range to compare (default EOT).

```
ficacheck usage: ficacheck [options] <FICA file>
ficcheck usage:  ficcheck [options] <FIC file>
```

3.17.3 Examples

```
>ficcheck fic06.187

Checking fic06.187
Read 252 records.


> ficacheck brokenfica

Checking brokenfica
text 0:Bad block header, record=2 location=484
text 1:blkHdr=[    ]
text 2:In record 2
text 3:In file brokenfica
text 4:Near file line 10
location 0:src/FICData.cpp:963
location 1:src/FFStream.cpp:159
location 2:src/FFStream.hpp:208
location 3:src/FFStream.hpp:208
```

3.17.4 Notes

Only the first error in each file is reported. The entire file is always checked regardless of time options.

3.18 *ficafic ficfica*

3.18.1 Overview

These applications convert navigation message data between variations of the FIC format, a format for GPS observations established by ARL:UT. *ficacheck* works with ASCII FIC files and *ficcheck* works with binary FIC files.

3.18.2 Usage

ficafic ficfica

```
ficafic usage: ficafic <input fica file> <output fic file name>
ficfica usage: ficfica <input fic file> <output fica file name>
```

3.18.3 Examples

```
> ficfica fic06.187 fica06.187

sh: ficfica: not found
```

```
File Snippets

Binary FIC File

0000000
*
0000020                                    B   L   K       m  \0  \0  \0
0000030  \0  \0  \0  \0      \0  \0  \0  \0  \0  \0  \0   f 005  \0  \0
0000040 022  \0  \0  \0   >   f 301   " 260   i   {   !   f  \0   d 026
0000050 335 344   8  \t 002   b   C 035 205   7   4 027 241 372 210 006
0000060 006   }   Y   / 301 374   ?  \0   \   S 021   8   >   f 301   "
 . . .

ASCII FIC File

BLK    109    0   32    0
        1382          18   583099966   561736112   375652454   154723549
   490955266   389298053   109640353   794393862     4193473   940659548
   583099966   561744492   792779231   218793822   800301952    12009725
   793943984    14182503    56922219   427630416   583099966   561753060
  1073203199   309077037     1329639    15188054   182084772   733918588
  1072216082   792738524
BLK      9   60    0    0
 .13900000000000D+03 .35800000000000D+03 .41142600000000D+06 .10000000000000D+01
 .10000000000000D+01 .13820000000000D+04 .10000000000000D+01 .00000000000000D+00
 .00000000000000D+00 .91136000000000D+06 .00000000000000D+00-.10244548320770D-07
 .41760000000000D+06 .00000000000000D+00-.14779288903810D-11-.24207541719079D-03
 .00000000000000D+00 .00000000000000D+00 .00000000000000D+00 .18000000000000D+02
 . . .
```

3.19 *ficdiff*

3.19.1 Overview

The application compares the contents of two FIC files containing ephemeris data.

3.19.2 Usage

ficdiff

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-t	–time=TIME	Start of time range to compare (default BOT).
-e	–end-time=TIME	End of time range to compare (default EOT).

```
ephdiff usage: ficdiff [options] fic1 fic2
```

3.19.3 Examples

```
> ficdiff -t "08/01/2006 12:00:00" fic1 fic2
<FIC BlockNumber: 9
 floats:  139 362 172806 1 1 1386 1 0 0 55296 0 -4.19095e-09 180000 0 . . .
 integers:
 chars:

<FIC BlockNumber: 9
 floats:  139 362 172806 1 1 1386 1 0 0 59392 0 -6.98492e-09 179984 0 . . .
 integers:
 chars:
 . . .
```

3.20 *findMoreThan12*

3.20.1 Overview

This application finds when there are simultaneously more than 12 SVs above a given elevation.

3.20.2 Usage

findMoreThan12

Required Arguments

Short Arg.	Long Arg.	Description
-e	–eph-files=ARG	Ephemeris source file(s). Can be RINEX nav, SP3, or FIC.
-p	–position=ARG	Antenna position in ECEF (x,y,z) coordinates. Format as a string: "X Y Z".
-m	–min-elev=NUM	Give an integer for the elevation (degrees) above which you want to find more than 12 SVs at a given time.

Optional Arguments

Short Arg.	Long Arg.	Description
-h	–help	Print help usage.
-v	–verbose	Increase verbosity.

3.21 *IonoBias*

3.21.1 Overview

The application will open and read several preprocessed RINEX observation files (containing obs types EL, LA, LO, SR or SS) and use the data to estimate satellite and receiver biases and to compute a simple ionospheric model using least squares and the slant TEC values.

3.21.2 Usage

IonoBias

Required Arguments

Short Arg.	Long Arg.	Description
	–input	Input RINEX obs file name(s).

Optional Arguments

Short Arg.	Long Arg.	Description
-f		File containing more options
	–inputdir	Path for input file(s).

Ephemeris Input

Short Arg.	Long Arg.	Description
	–navdir	Path of navigation file(s).
	–nav	Navigation (RINEX (nav) OR SP3) file(s).

Output

Short Arg.	Long Arg.	Description
	–datafile	Data (AT) file name, for output and/or input.
	–log	Output log file name.
	–biasout	Output satellite+receiver biases file name.

Time Limits

Short Arg.	Long Arg.	Description
	–BeginTime	Start time, arg is of the form YYYY,MM,DD,HH,Min,Sec.
	–BeginGPSTime	Start time, arg is of the form GPSweek,GPSsow.
	–EndTime	End time, arg is of the form YYYY,MM,DD,HH,Min,Sec.
	–EndGPSTime	End time, arg is of the form GPSweek,GPSsow.

Processing

Short Arg.	Long Arg.	Description
	–NoEstimation	Do NOT perform the estimation (default=false).
	–NoPreprocess	Skip preprocessing; read (existing) AT file (false).
	–NoSatBiases	Compute Receiver biases ONLY (not Rx+Sat biases) (false).
	–Model	Ionospheric model: type is linear, quadratic or cubic.
	–MinPoints	Minimum points per satellite required.
	–MinTimeSpan	Minimum timespan per satellite required (minutes).
	–MinElevation	Minimum elevation angle (degrees).

–MinLatitude	Minimum latitude (degrees).
–MaxLatitude	Maximum latitude (degrees).
–MinLongitude	Minimum longitude (degrees).
–MaxLongitude	Maximum longitude (degrees).
–TimeSector	Time sector (day — night — both).
–TerminOffset	Terminator offset (minutes).
–IonoHeight	Ionosphere height (km).

Other Options

Short Arg.	Long Arg.	Description
	–XSat	Exclude this satellite (<sat> may be <system> only).
-v	–verbose	Print extended output info.
-d	–debug	Increase debug level.
-h	–help	Print syntax and quit.

3.21.3 Examples

```
> IonoBias --inputdir data_set --navdir data_set --input s081213a.99o --input s081214a.99o
--input s081215a.99o --nav s081213a.99n --nav s081214a.99n --nav s081215a.99n --datafile output}
IonoBias, built on the GPSTK ToolKit, Ver 1.0 6/25/04, Run 2006/08/17 09:50:59
IonoBias output directed to log file IonoBias.log
IonoBias timing: 6.210 seconds.
```

```
Output File Snippet

    3     3 Number (max, good) stations in this file
01010110110000111111011101110111 0
010101101100001111110111011101110
010100101100001111110111011101110
Npt  9737 Sta 85408 LLH    30.2160   262.2746   163.4226
1021      0.0   0.00000 -463513.64930 0.32    0.000      1  1   1
1021      0.0   0.00000 -463513.64930 0.32    0.000      1 14   1
1021      0.0   0.00000 -463513.64930 0.32    0.000      1 15   1
1021      0.0   0.00000 -463513.64930 0.32    0.000      1 21   1
1021      0.0   0.00000 -463513.64930 0.32    0.000      1 22   1
1021      0.0   0.00000 -463513.64930 0.32    0.000      1 25   1
1021      0.0   0.00000 -463513.64930 0.32    0.000      1 29   1
1021      0.0   0.00000 -463513.64930 0.32    0.000      1 30   1
1021     30.0   0.00000 -463513.52430 0.32    0.000      1  1   1
1021     30.0   0.00000 -463513.52430 0.32    0.000      1 14   1
```

3.21.4 Notes

Input can be either on the command line or put in a file and then input using the -f option. The file is formatted just as if it were the command line.

3.22 *mdp2fic mdp2rinex*

3.22.1 Overview

The applications convert a variety of GPS related observations from the MDP format to FIC and RINEX formats. MDP is a format for network receiver interfaces derived by ARL:UT that can be used to serve observations over networks.

3.22.2 Usage

mdp2fic mdp2rinex

Required Arguments

Short Arg.	Long Arg.	Description
-i	–mdp-input=ARG	Filename to read MDP data from. The filename of '-' means to use stdin.
-n	–nav=ARG	Filename to which FIC nav data will be written.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-l	–log=ARG	Filename for (optional) output log file.

mdp2rinex

Required Arguments

Short Arg.	Long Arg.	Description
-i	–mdp-input=ARG	Filename to read MDP data from. The filename of '-' means to use stdin.
-n	–obs=ARG	Filename to write RINEX obs data to. The filename of '-' means to use stdout.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-n	–nav=ARG	Filename to write RINEX nav data.
-p	–pos=ARG	Antenna position to write into obs file header. Format as string: "X Y Z"
-t	–thinning=ARG	A thinning factor for the data, specified in seconds between points.
-c	–12c	Enable output of L2C data in C2.
-a	–any-nav-source	Accept subframes from any code/carrier.

3.22.3 Examples

```
> mdp2fic -i mdp183.06 -o fic183.06 -l mdp2ficlog183.06
```

```
> mdp2rinex -i mdp183.06 -o rin183.06o -n rin183.06n -t 60
```

3.23 *mdptool*

3.23.1 Overview

The application performs various functions on a stream of MDP data.

3.23.2 Usage

mdptool

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-i	–input=ARG	Where to get the MDP data from. The default is to use stdin. If the file name begins with "tcp:" the remainder is assumed to be a hostname[:port] and the source is taken from a tcp socket at this address. If the port number is not specified a default of 8910 is used.
	–output=ARG	Where to send the output. The default is stdout.
-p	–pvt	Enable pvt output.
-o	–obs	Enable obs output.
-n	–nav	Enable nav output.
-t	–test	Enable selftest output.
-x	–hex	Dump all messages in hex.
-b	–bad	Try to process bad messages also.
-a	–almanac	Build and process almanacs. Only applies to the nav style.
-e	–ephemeris	Build and process engineering ephemerides. Only applies to the nav style.
	–min-alm	This allows a complete almanac to be constructed from fewer than 50 pages. It is required for Ashtech Z(Y)-12. The default is to require all 50 pages.
-f	–follow	Follow the input file as it grows.
-s	–output-style=ARG	What type of output to produce from the MDP stream. Valid styles are: brief, verbose, table, track, null, mdp, nav, and summary. The default is summary. Some modes aren't quite complete.
-l	–timeSpan=NUM	How much data to process, in seconds.
-m	–bug-mask=NUM	What RX bugs: 1 SV count, 2 nav parity/fmt, 4 HOW/hdr time equal.
	–startTime=TIME	Ignore data before this time. (%4Y/%03j/%02H:%02M:%05.2f).
	–stopTime=TIME	Ignore any data after this time.
	–time-format=ARG	CommonTime format specifier used for times in the output. The default is %4Y %3j %02H:%02M:%04.1f.

3.23.3 Examples

```
> mdptool -i mdp/85408-2012131-2u.mdp -a
Done processing data.


Header summary:
  Processed 2685 headers.
  First freshness count was d96a
  Last freshness count was  e3e6
  Encountered 0 breaks in the freshness count

Observation Epoch message summary:
No Observation Epoch messages processed.

PVT Solution message summary:
  Pvt data spans 2012/131/00:02:06.0 to 2012/131/17:57:51.0 (17:55:45.0)
  PVT output rate is 1.5 sec.


Navigation Subframe message summary:
  No Navigation Subframe messages processed.
```

3.23.4 Notes

In the summary mode, the default is to only summarize the observation data above 10 degrees. Increasing the verbosity level will also summarize the data below 10 degrees.

3.24 *mergeFic*

3.24.1 Overview

This application merges multiple FIC files into a single FIC file.

3.24.2 Usage

mergeFic

Required Arguments		
Short Arg.	Long Arg.	Description
-i	–input=ARG	An input FIC observation file, can be repeated as many times as needed.
-o	–output=ARG	Name for the merged output FIC observation file. Any existing file with that name will be overwritten.
Optional Arguments		
Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.

3.24.3 Examples

```
> mergeFIC -i fic1 -i fic2 -o ficm
```

3.25 *mergeRinObs mergeRinNav mergeRinMet*

3.25.1 Overview

These applications merge multiple RINEX observation, navigation, or meteorological data files into a single coherent RINEX obs/nav/met file, respectively.

3.25.2 Usage

mergeRinObs mergeRinNav mergeRinMet

Required Arguments		
Short Arg.	Long Arg.	Description
-i	–input=ARG	An input RINEX observation file, can be repeated as many times as needed.
-o	–output=ARG	Name for the merged output RINEX observation file. Any existing file with that name will be overwritten.
Optional Arguments		
Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.

mergeRinNav and *mergeRinMet* have the same usage.

3.25.3 Examples

```
> mergeRinObs -i arl280.06o -i arl2810.06o -o arl280-10.06o

> mergeRinNav -i arl280.06n -i arl2810.06n -o arl280-10.06n

> mergeRinMet -i arl280.06m -i arl2810.06m -o arl280-10.06m
```

3.26 *navdmp*

3.26.1 Overview

The application prints the contents of an FIC or RINEX navigation file into a human readable file and allows filtering of the data.

3.26.2 Usage

navdmp

Required Arguments

Short Arg.	Long Arg.	Description
-i	–input=ARG	Name of an input navigation message file.
-o	–output=ARG	Name of an output file.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-a	–all-records	Unless otherwise specified, use default values for record filtration.
-t	–time=TIME	Start time (of data) for processing.
-e	–end-time=TIME	End time (of data) for processing.
-p	–prn=NUM	PRN(s) to include.
-b	–block=NUM	FIC block number(s) to process ((9)109 (Engineering) ephemerides, (62)162 (engineering) almanacs).
-r	–RINEX	Assume input file is a RINEX navigation message file.

3.26.3 Examples

```
> navdmp -i data_set/s081213a.99n -o summary --RINEX

Current filtering options:
       Start time:   01/06/1980 00:00:00
       End time:     01/01/4713 00:00:00
       PRNs:         using all PRNs

Choose an option by number then push enter:
       1) Change the start time
       2) Change the end time
       3) Select specific PRNs
       5) Process the file
use ctrl-c to exit
? 5

processing...
```

```
Summary File Snippet

****************************************************************************
```

```
Broadcast Ephemeris (Engineering Units)

PRN : 14

             Week(10bt)     SOW     DOW   UTD     SOD   MM/DD/YYYY   HH:MM:SS
Clock Epoch:  1021(1021)    7200   Sun-0   213    7200   08/01/1999   02:00:00
Eph Epoch:    1021(1021)    7200   Sun-0   213    7200   08/01/1999   02:00:00
Transmit Week:1021
Fit interval flag :  0

          SUBFRAME OVERHEAD

              SOW    DOW:HH:MM:SS     IOD    ALERT   A-S
SF1 HOW:        6  Sun-0:00:00:06   0x023      0     off
SF2 HOW:        6  Sun-0:00:00:06    0x23      0     off
SF3 HOW:        6  Sun-0:00:00:06    0x23      0     off

           CLOCK

Bias T0:       2.82567926E-05 sec
Drift:         1.02318154E-12 sec/sec
Drift rate:    0.00000000E+00 sec/(sec**2)
Group delay:  -2.32830644E-09 sec

           ORBIT PARAMETERS

Semi-major axis:         5.15359685E+03 m**.5
Motion correction:       4.44732811E-09 rad/sec
Eccentricity:            8.10711295E-04
Arg of perigee:          2.16661714E+00 rad
Mean anomaly at epoch:   1.75307843E-01 rad
Right ascension:         2.02857661E+00 rad     -8.31963226E-09 rad/sec
Inclination:             9.77089255E-01 rad      2.20723480E-10 rad/sec

           HARMONIC CORRECTIONS

Radial        Sine:   1.31875000E+01 m    Cosine:   3.31593750E+02 m
Inclination   Sine:   5.77419996E-08 rad  Cosine:  -1.86264515E-08 rad
In-track      Sine:   2.74367630E-06 rad  Cosine:   6.27711415E-07 rad

           SV STATUS

Health bits:   0x00      URA index:    7
Code on L2:    P only      L2 P Nav data:          on

****************************************************************************
```

3.27 *NavMerge*

3.27.1 Overview

The application merges RINEX navigation files into a single file.

3.27.2 Usage

NavMerge

Optional Arguments

Short Arg.	Long Arg.	Description
-o		Write all data to an output RINEX nav file. If omitted, a data summary is written to the screen.
-tb		Output only if epoch is within 4 hours of the interval (tb,te).
-te		If te or tb is missing, they are made equal. Time tags have the form year,mon,day,HH,min,sec OR GPSweek,sow.

```
NavMerge usage: NavMerge [options] <RINEX nav file> <RINEX nav file>
```

3.27.3 Examples

```
> NavMerge -o s081213-214.99n s081213a.99n s081214a.99n

Output file name is
Exception: text 0:Unexpected EOF
text 1:In record 0
text 2:In file s081213-214.99n
text 3:Near file line 0
location 0:src/FFTextStream.hpp:244
location 1:src/FFStream.cpp:159
location 2:src/FFStream.hpp:208
location 3:src/FFStream.hpp:208

Read    0 ephemerides from file s081213-214.99n
Read  200 ephemerides from file s081213a.99n
Read  197 ephemerides from file s081214a.99n
Read  397 total ephemerides.
```

3.27.4 Notes

NavMerge corrects data for output when the GPS full week number is inconsistent with the epoch time.

3.28 *navsum*

3.28.1 Overview

This application lists the block contents of a FIC file and prints summary count information.

3.28.2 Usage

navsum

Required Arguments

Short Arg.	Long Arg.	Description
-i	–input=ARG	Name of an input FIC file.
-o	–output=ARG	Name of an output file.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-a	–all-records	Unless otherwise specified, use default values for record filtration.
-t	–time=TIME	Start time (of data) for processing.
-e	–end-time=TIME	End time (of data) for processing.
-p	–prn=NUM	PRN(s) to include.
-b	–block=NUM	FIC block number(s) to process ((9)109 (Engineering) ephemerides, (62)162 (engineering) almanacs).
-f	–use-alternate-format	Use alternate output format.

3.28.3 Examples

```
> navsum -i s081213a.99n -o summary --RINEX

Current filtering options:
       Start time:   01/06/1980 00:00:00
       End time:     01/01/4713 00:00:00
       PRNs:         using all PRNs

Choose an option by number then push enter:
       1) Change the start time
       2) Change the end time
       3) Select specific PRNs
       5) Process the file
use ctrl-c to exit
? 5

processing...
```

```
Summary of data processed
Block Type Summary
Type   # Blocks Found
   9             0
 109             0
```

```
  62              0
 162              0

Ephemeris Blocks by PRN
PRN Block      Num
 01     9          0
 01   109          0
 02     9          0
 02   109          0
 03     9          0
 03   109          0
 04     9          0
 04   109          0
 05     9          0
 . . .
```

3.29 *novaRinex*

3.29.1 Overview

The application will open and read a binary Novatel file (OEM2 and OEM4 receivers are supported), and convert the data to RINEX format observation and navigation files. The RINEX header is filled using user input (see below), and optional records are filled.

3.29.2 Usage

novaRinex

Required Arguments

Short Arg.	Long Arg.	Description
	–input	Novatel binary input file.

Optional Arguments

Short Arg.	Long Arg.	Description
-f	–file	Name of file containing more options (ignores '#' to EOL).
	–dir	Directory in which to find input file (default ./).
	–obs	RINEX observation output file (RnovaRINEX.obs).
	–nav	RINEX navigation output file (RnovaRINEX.nav).

Output RINEX Header Fields

Short Arg.	Long Arg.	Description
	–noHDopt	If present, do not fill optional records in the output RINEX header.
	–HDp	Set output RINEX header 'program' field ('novaRINEX v2.1 9/07').
	–HDr	Set output RINEX header 'run by' field ('ARL:UT/GPSTk').
	–HDo <obser>	Set output RINEX header 'observer' field.
	–HDa <agency>	Set output RINEX header 'agency' field ('ARL:UT/GPSTk').
	–HDm <marker>	Set output RINEX header 'marker' field.
	–HDn <number>	Set output RINEX header 'number' field.
	–HDrn <number>	Set output RINEX header 'Rx number' field.
	–HDrt <type>	Set output RINEX header 'Rx type' field ('Novatel').
	–HDrv <vers>	Set output RINEX header 'Rx version' field ('OEM2/4').
	–HDan <number>	Set output RINEX header 'antenna number' field.
	–HDat <type>	Set output RINEX header 'antenna type' field.
	–HDc <comment>	Add comment to output RINEX header (>1 allowed).

Output RINEX Observation Data

Short Arg.	Long Arg.	Description
	–obstype <OT>	Output this RINEX (standard) obs type (i.e. <OT> is one of L1,L2,C1,P1,P2,D1,D2,S1,or S2); repeat for each type. NB default is ALL std. types that have data.

Output Configuration

Short Arg.	Long Arg.	Description
	–begin <arg>	Start time, arg is of the form YYYY,MM,DD,HH,Min,Sec.
	–beginGPS <arg>	Start time, arg is of the form GPSweek,GPSsow.
	–end <arg>	End time, arg is of the form YYYY,MM,DD,HH,Min,Sec.
	–endGPS <arg>	End time, arg is of the form GPSweek,GPSsow
	–week <week>	GPS Week number of this data, NB: this is for OEM2; this command serves two functions, resolving the ambiguity in the 10-bit week (default uses –begin, –end, or the current system time) and ensuring that ephemeris records that precede any obs records are not lost.
	–debias	Remove an initial bias from the phase.
-h	–help	Print this message and quit.
	–verbose	Print more information.
-d	–debug	Print extended output info.

3.29.3 Notes

Input is on the command line, or of the same format in a file (-f<file>).

3.30 *poscvt*

3.30.1 Overview

This application allows the user to convert among different coordinate systems on the command line. Coordinate systems handled include Cartesian, geocentric, and geodetic.

3.30.2 Usage

poscvt

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
	–ecef=POSITION	ECEF "X Y Z" in meters.
	–geodetic=POSITION	Geodetic "lat lon alt" in deg, deg, meters.
	–geocentric=POSITION	Geocentric "lat lon radius" in deg, deg, meters.
	–spherical=POSITION	Spherical "theta, pi, radius" in deg, deg, meters.
-l	–list-formats	List the available format codes for use by the input and output format options.
-F	–output-format=ARG	Write the position with the given format.

3.30.3 Examples

```
> poscvt --ecef="4345070.59253 45619878.26297 803.598856837"

    ECEF (x,y,z) in meters              4345070.5925 45619878.2630 803.5989
    Geodetic (llh) in deg, deg, m       0.00100566 84.55926933 39448197.4795
    Geocentric (llr) in deg, deg, m     0.00100472 84.55926933 45826334.4795
    Spherical (tpr) in deg, deg, m      89.99899528 84.55926933 45826334.4795
```

3.30.4 Notes

If no options are given *poscvt* assumes XYZ 0 0 0.

3.31 *PRSolve*

3.31.1 Overview

The application reads one or more RINEX observation files, plus one or more navigation (ephemeris) files, and computes an autonomous pseudorange position solution, using a RAIM-like algorithm to eliminate outliers. Output is to the log file, and also optionally to a RINEX observation file with the position solutions in auxiliary header blocks.

3.31.2 Usage

PRSolve

Required Arguments

Short Arg.	Long Arg.	Description
-o	–obs	Input RINEX observation file(s).
-n	–nav	Input navigation (ephemeris) file(s) (RINEX or SP3).

Optional Arguments: Input

Short Arg.	Long Arg.	Description
-f		File containing more options.
	–obsdir	Directory of input observation file(s).
	–navdir	Directory of input navigation file(s).
	–metdir	Directory of input meteorological file(s).
-m	–met	Input RINEX meteorological file(s).
	–decimate	Decimate data to time interval dt.
	–BeginTime	Start time: arg is 'GPSweek,sow' OR 'YYYY,MM,DD,HH,Min,Sec'.
	–EndTime	End time: arg is 'GPSweek,sow' OR 'YYYY,MM,DD,HH,Min,Sec'.
	–useCA	Use C/A code pseudorange if P1 is not available.
	–forceCA	Use C/A code pseudorange regardless of P1 availability.

Optional Arguments: Configuration

Short Arg.	Long Arg.	Description
	–Freq	Frequency to process: 1, 2, or 3 for L1, L2, or iono-free combination.
	–MinElev	Minimum elevation angle in degrees (only if –PosXYZ).
	–exSat	Exclude this satellite.
	–Trop	Trop model, one of ZR, BL, SA, NB, NL, GG, GGH (gpstk::TropModel), with optional weather T(c), P(mb),RH(%).

Optional Arguments: PRSolution Configuration

Short Arg.	Long Arg.	Description
	–RMSlimit	Upper limit on RMS post-fit residuals (m) for a good solution.
	–SlopeLimit	Upper limit on RAIM 'slope' for a good solution.
	–Algebra	Use algebraic algorithm (otherwise linearized LS).
	–DistanceCriterion	Use distance from a priori as convergence criterion (else RMS).
	–ReturnAtOnce	Return as soon as a good solution is found.

	–NReject	Maximum number of satellites to reject.
	–NIter	Maximum iteration count (linearized LS algorithm).
	–Conv	Minimum convergence criterion (m) (LLS algorithm).

Optional Arguments: Output

Short Arg.	Long Arg.	Description
	–Log	Output log file name (prs.log).
	–PosXYZ <X,Y,Z>	Known position (ECEF,m), used to compute output residuals.
	–APSout	Output autonomous pseudorange solution (APS - no RAIM).
	–TimeFormat	Output time format (ala CommonTime) (default: %4F %10.3g).

Optional Arguments: RINEX Output

Short Arg.	Long Arg.	Description
	–outRinex	Output RINEX observation file name.
	–RunBy	Output RINEX header 'RUN BY' string.
	–Observer	Output RINEX header 'OBSERVER' string.
	–Agency	Output RINEX header 'AGENCY' string.
	–Marker	Output RINEX header 'MARKER' string.
	–Number	Output RINEX header 'NUMBER' string.

Optional Arguments: Help

Short Arg.	Long Arg.	Description
	–verbose	Print extended output.
	–debug	Print very extended output.
	–helpRetCodes	Print return codes (implies –help).
-h	–help	Print syntax and quit.

3.31.3 Examples

```
> PRSolve -o arl2800.06o -n arl2800.06n

PRSolve, part of the GPS ToolKit, Ver 2.3 11/09, Run 2011/07/22 11:39:15
Opened log file prs.log

Weighted average RAIM solution for file: arl2800.06o
 (2880 total epochs, with 2880 good, 0 rejected.)
    918129.266960  -4346070.850055   4561977.615781
Covariance of RAIM solution for file: arl2800.06o
         0.000150        -0.000061         0.000058
        -0.000061         0.000427        -0.000248
         0.000058        -0.000248         0.000493
```

3.31.4 Notes

In the log file, results appear one epoch per line with the format:
TAG Nrej week sow Nsat X Y Z T RMS slope nit conv sat sat .. (code) [N]V
TAG denotes solution (X Y Z T) type:

- RPF Final RAIM ECEF XYZ solution

- RPR Final RAIM ECEF XYZ solution residuals [only if –PosXYZ given]
- RNE Final RAIM North-East-Up solution residuals [only if –PosXYZ]
- APS Autonomous ECEF XYZ solution [only if –APSout given]
- APR Autonomous ECEF XYZ solution residuals [only if both –APS & –Pos]
- ANE Autonomous North-East-Up solution residuals [only if –APS & –Pos]

Where:

- Nrej = number of rejected sats
- (week,sow) = GPS time tag
- Nsat = # sats used
- XYZT = position+time solution(or residuals)
- RMS = RMS residual of fit
- slope = RAIM slope
- nit = # of iterations
- conv = convergence factor
- ‘sat sat ...’ lists all sat. PRNs (- : rejected)
- code = return value from PRSolution::RAIMCompute()
- NV means NOT valid

3.32 *ResCor*

3.32.1 Overview

The application will open and read a single RINEX observation file, apply editing commands using the RinexEditor package, compute any of several residuals and corrections and register extended RINEX observation types for them, and then write the edited data, along with the new extended observation types, to an output RINEX observation file.

NOTE: ResCor is only available in GPSTK 1.x. It is only compatiable with Rinex versions 2.1 and earlier

3.32.2 Usage

ResCor

Required Arguments

Short Arg.	Long Arg.	Description
-IF		Input RINEX observation file.
-OF		Name of ouput RINEX observation file.

Configuration Arguments

Short Arg.	Long Arg.	Description
-f<file>		File containing more options.
	–nav <file>	Navigation (RINEX Nav OR SP3) file(s).
	–navdir <dir>	Directory of navigation file(s).

Reference Position Input

Short Arg.	Long Arg.	Description
	–RxLLH <l,l,h>	1.Receiver position (static) in geodetic lat, lon(E), ht (deg,deg,m).
	–RxXYZ <x,y,z>	2.Receiver position (static) in ECEF coordinates (m).
	–Rxhere	3.Reference site positions(time) from this file (i.e. -IF<RINEXFile>).
	–RxRinex <fn>	4.Reference site positions(time) from another RINEX file named <fn>.
	–RxFlat <fn>	5.Reference site positions and times given in a flat file named <fn>.
	–Rxhelp	(Enter –Rxhelp for a description of the -RxFlat file format).
	–RAIM	6.Reference site positions computed via RAIM (requires P1,P2,EP). NB the following two options apply only if –RAIM is found.
	–noRAIMedit	Do not edit data based on RAIM solution.
	–RAIMhead	Output average RAIM solution to RINEX header (if -HDf also appears).
	–noRefout	Do not output reference solution to RINEX.
	–MinElev	Minimum satellite elevation in degrees for output.

Residual/Correction Computation

Short Arg.	Long Arg.	Description
	–debias <OT,l>	Debias new output type <OT>; trigger a bias reset with limit <l>.

	–Callow	Allow C1 to replace P1 when P1 is not available.
	–Cforce	Force C/A code pseudorange C1 to replace P1.
	–IonoHt <ht>	Height of ionosphere in km (default 400) (needed for LA,LO,VR,VP).
	–Tgd	Apply the Tgd from BC ephemeris to SR,SP,VR, and VP.
	–SVonly <prn>	Process this satellite ONLY.

Output Files

Short Arg.	Long Arg.	Description
	–Log <file>	Output log file name (rc.log)

Help

Short Arg.	Long Arg.	Description
	–verbose	Print extended output
	–debug	Print debugging information.
-h	–help	Print syntax and quit.
	–REChelp	Print syntax of RINEXEditor commands and quit.
	–ROThelp	Print list of extended RINEX observation types and quit.

List of Available RINEX Observation Types

```
OT Description          Units     Required input (EP=ephemeris,PS=Rx Position)
ER Ephemeris range      meters                 EP PS
RI Iono Delay, Range    meters              P1
PI Iono Delay, Phase    meters     L1 L2
TR Tropospheric Delay   meters                 EP PS
RL Relativity Correct.  meters                 EP
SC SV Clock Bias        meters                 EP
EL Elevation Angle      degrees                EP PS
AZ Azimuth Angle        degrees                EP PS
SR Slant TEC (PR)       TECU                P1
SP Slant TEC (Ph)       TECU       L1 L2
VR Vertical TEC (PR)    TECU                P1 EP PS
VP Vertical TEC (Ph)    TECU       L1 L2       EP PS
LA Lat Iono Intercept   degrees                EP PS
LO Lon Iono Intercept   degrees                EP PS
P3 TFC(IF) Pseudorange  meters              P1
L3 TFC(IF) Phase        meters     L1 L2
P4 GeoFree Pseudorange  meters              P1
L4 GeoFree Phase        meters     L1 L2
P5 WideLane Pseudorange meters              P1
L5 WideLane Phase       meters     L1 L2
MP Multipath (=M3)      meters     L1 L2    P1
M1 L1 Range minus Phase meters     L1       P1
M2 L2 Range minus Phase meters        L2
M3 IF Range minus Phase meters     L1 L2    P1
M4 GF Range minus Phase meters     L1 L2    P1
M5 WL Range minus Phase meters     L1 L2    P1
XR Non-dispersive Range meters     L1 L2    P1
XI Ionospheric delay    meters     L1 L2    P1
X1 Range Error L1       meters     L1 L2    P1
X2 Range Error L2       meters     L1 L2    P1
SX Satellite ECEF-X     meters                 EP
```

```
SY Satellite ECEF-Y     meters                 EP
SZ Satellite ECEF-Z     meters                 EP
```

3.33 *reszilla*

3.33.1 Overview

Reszilla is a set of applications that compute various residuals from GPS pseudorange, phase, and doppler data. These data are often refered to as raw observations. The two types of residuals that are currently computed are an Observed Range Deviation (ORD), and a double difference (DD). Once these residuals are computed, statistical summaries of these differences are computed and output to the user. Optionally, the residuals themselves may be output.

3.33.2 Observed Range Deviations

An ORD is basically the observed range to an SV differenced from the estimated range to that SV. There are many terms that go into computing the estimated range and/or correcting the observed range for known effects. When all of these effects are accounted for (as reszilla is capable of doing) ORDs can be in the 10-30 cm range for a geodetic quality GPS receiver. Pretty impressive when you consider that the range to the SV is somewhere between 20 to 26 million meters.

For many GPS receivers, the most significant effect to account for is the receiver clock offset. This is the difference between the receivers internal time and true GPS time. This parameter is often computed as part of a PVT solution. This is not how reszilla works. Reszilla is provided a surveyed position of the receiver antenna, and it makes a more accurate estimate of the receiver clock offset by averaging the residuals of all SVs in track.

3.33.3 Usage

OrdApp

Required Arguments

Short Arg.	Long Arg.	Description
-i	–input	Where to read the ord data. The default is stdin.
-r	–output	Where to write the output. The default is stdout.
-t	–time-format	CommonTime format specifier used for times in the output.

Optional Arguments

Short Arg.	Long Arg.	Description
	–ns	Report the clock in ns, not meters.

ordClock

ordClock generates clock estimates for each epoch of ORDs.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.

-h	–help	Print help usage.
-w	–use-warts	Use warts in the clock solution. The default is to not use warts.
-e	–estimate-only	Only compute the receiver clock bias. Don't remove this bias from the ords. The default is to both estimate the bias and remove the it from the ords.
-c	–clock-source=ARG	An ord file to read the receiver clock offsets from.
-i	–input=ARG	Where to read the ord data. The default is stdin.
-r	–output=ARG	Where to write the output. The default is stdout.
-t	–time-format=ARG	CommonTime format specifier used for times in the output. The default is "%4Y %3j %02H:%02M:%04.1f".
	–ns	Report the clock in ns, not meters.

Examples

```
> ordClock -i ords.out
# Time              Type PRN  Elev   Azimuth       ORD(m) wonky
2004 162 00:00:00.0    0   0   0.00    0.00        0.95000    67
2004 162 00:00:00.0    1                  0.95000     1
```

ordEdit

ordEdit edits an ORD file based on various criteria.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-k	–clock-est	Remove ORDs that do not have corresponding clock estimates.
-c	–no-clock	Remove all clock offset estimate warts. Give this option twice to remove all clock data.
-m	–elev=NUM	Remove data for SVs below a given elevation mask.
-p	–PRN=NUM	Filter data by PRN number. Repeat option for multiple satellites. Negative PRN numbers mean exclude these PRNs. Positive PRN numbers mean only include these satellites. Zero removes all.
-w	–warts=NUM	Include/Exclude warts from the indicated PRN. Repeat option for multiple PRNs. Negative numbers exclude, positive numbers include, zero excludes warts from all PRNs. The default is to include all warts.
-e	–be-file=ARG	Remove data for unhealthy SVs by providing broadcast ephemeris source: RINEX nav or FIC file.

	–start=ARG	Throw out data before this time. Format as string: "yyyy ddd HH:MM:SS".
	–end=ARG	Throw out data after this time. Format as string: "yyyy ddd HH:MM:SS".
-s	–size=ARG	Remove clock residuals with absolute values greater than this size (meters).
-l	–ord-limit=ARG	Remove ords with absolute values greater than this size (meters).
-i	–input=ARG	Where to read the ord data. The default is stdin.
-r	–output=ARG	Where to write the output. The default is stdout.
-t	–time-format=ARG	CommonTime format specifier used for times in the output. The default is "%4Y %3j %02H:%02M:%04.1f".
	–ns	Report the clock in ns, not meters.

Examples

```
> ordEdit -i ords.out

# Time              Type PRN  Elev   Azimuth       ORD(m) wonky
2004 162 00:00:00.0    0   0   0.00    0.00        0.95000    67
2004 162 00:00:00.0    1                   0.95000     0
```

ordGen

ordGen generates observed range deviations.

Required Arguments

Short Arg.	Long Arg.	Description
-o	–obs=ARG	Where to get the obs data.
-e	–eph=ARG	Where to get the ephemeris data. Acceptable formats include RINEX (nav), FIC, MDP, SP3, YUMA, and SEM.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-w	–weather=ARG	Weather data file name (RINEX met format only).
-c	–msc=ARG	Station coordinate file.
	–omode=ARG	Specifies what observations are used to compute the ORDs. Valid values are:p1p2, z1z2, c1p2, c1c2, c1y2, c1z2, y1y2, c1, p1, y1, z1, c2, p2, y2, z2, smo, dynamic, and smart. The default is smart.
	–trop-model=ARG	Specify the trop model to use. Options are zero, simple, nb, and gg. The default is nb.
-p	–pos=ARG	Location of the antenna in meters ECEF.
-m	–msid=NUM	Station to process data for. Used to select a station position from the msc file or data from a SMODF file.
-n	–near	Allows the program to select an ephemeris that is not strictly in the future. Only affects the selection of which broadcast ephemeris to use.

	–sv-time	Assume that the data is time-tagged according to each SV's clock, not a common receiver clock. The is set by default only for omode=smo.
-i	–input=ARG	Where to read the ord data. The default is stdin.
-r	–output=ARG	Where to write the output. The default is stdout.
-t	–time-format=ARG	CommonTime format specifier used for times in the output. The default is "%4Y %3j %02H:%02M:%04.1f".
	–ns	Report the clock in ns, not meters.

Examples

```
> ordGen -o obs/s121001a.09o -e nav/s121001a.09n -r ordGen.out

-------------------------ordGen.out----------------------------

# Time              Type PRN  Elev   Azimuth       ORD(m) wonky
2009   1 00:00:30.0    0   2  37.95  236.83   269828.15589     0
2009   1 00:00:30.0    0   4  53.67  179.46   269828.54600     0
2009   1 00:00:30.0    0   5   7.76  224.54   269829.73261     0
2009   1 00:00:30.0    0  10  27.65  317.24   269828.24818     0
2009   1 00:00:30.0    0  12  18.86  230.26   269825.94699     0
2009   1 00:00:30.0    0  13  29.10   96.35   269827.83224     0
2009   1 00:00:30.0    0  17  57.26   72.48   269827.82792     0
2009   1 00:00:30.0    0  23  17.26  122.49   269826.61070     0
2009   1 00:00:30.0    0  28  15.47    6.38   269823.93863     0
2009   1 00:01:00.0    0   2  38.05  236.58   269826.49879     0
...
----------------------------------------------------------------
```

ordLinEst

ordLinEst computes a linear clock estimate.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-m	–max-rate=ARG	Rate used to detect a clock jump. Default is 10,000 m/day.
-i	–input=ARG	Where to read the ord data. The default is stdin.
-r	–output=ARG	Where to write the output. The default is stdout.
-t	–time-format=ARG	CommonTime format specifier used for times in the output. The default is "%4Y %3j %02H:%02M:%04.1f".
	–ns	Report the clock in ns, not meters.

Examples

```
> ordLinEst -i ords.out
#  t0                  t1                   t0 offset(m)  t1 offset(m)  slope(m/d)  abdev(m)
#  -------------------  -------------------  ------------  ------------  ----------  --------
#
# Time              Type PRN  Elev   Azimuth       ORD(m) wonky
2004 162 00:00:00.0    0   0   0.00    0.00        0.95000    67
2004 162 00:00:00.0    1                  0.95000     0
```

ordStats

ordStats computes ORD statistics.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-b	–elev-bin=ARG	A range of elevations, used in computing the statistical summaries. Repeat to specify multiple bins. The default is "-b 0-10 -b 10-20 -b 20-60 -b 10-90".
-s	–sigma=NUM	Multiplier for sigma stripping used in statistical computations. The default value is 6.
-w	–wonky	Use wonky data in stats computation. The default is to not use such data.
	–stats-only	Only output stats to stdout.
-i	–input=ARG	Where to read the ord data. The default is stdin.
-r	–output=ARG	Where to write the output. The default is stdout.
-t	–time-format=ARG	CommonTime format specifier used for times in the output. The default is "%4Y %3j %02H:%02M:%04.1f".
	–ns	Report the clock in ns, not meters.

3.33.4 Double Difference Residuals

While many double differences exist, reszilla computes the first difference to a master SV and the second difference to a second receiver. This double difference removes receiver clock error, iono, trop, and SV clock errors. When the two receivers are connected to a common antenna (often referred to as a zero-baseline setup) and are of the same type, even the multipath is differenced out. What is left is basically receiver tracking noise and receiver tracking errors.

One complicating factor in computing this DD is that while the clock errors in the receivers cancel out, there is still an error associated with the motion of the satellite during the interval between when the two receivers are computing their observation. To remove this error, an estimate of the clock offset between the two receivers is needed. Reszilla can get this estimate in one of two ways; estimate this by computing a clock estimate for each receiver as described under the ORD section or read the estimates from the rinex obs data files. These two estimates are then differenced to get the offset between the two receivers.

Another complicating factor is that the phase observations normally have an "integer ambiguity" associated with them. When the DD phase observation is computed, it will have the difference between the two receivers' ambiguity. Often this number can be quite big. Removing this ambiguity is often referred to as debiasing the data. This process involves much black magic and sleight of hand. Do not delve into this or even look too closely at the details or you will be sullied.

3.33.5 Usage

ddGen

ddGen computes double-difference residuals from raw observations.

Required Arguments

Short Arg.	Long Arg.	Description
-1	–obs1=ARG	Where to get the first receiver's obs data.
-2	–obs2=ARG	Where to get the second receiver's obs data.
-e	–eph=ARG	Where to get the ephemeris data. Acceptable formats include RINEX nav, FIC, MDP, SP3, YUMA, and SEM.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
	–ddmode=ARG	Specifies what observations are used to compute the double difference residuals. Valid values are: all, phase. The default is all.
	–omode=ARG	Specifies what observations to use to compute the ORDs. Valid values are: p1p2, z1z2, c1p2, c1y2, c1z2, y1y2, c1, p1, y1, z1, c2, p2, y2, z2 smo, dynamic, and smart. The default is smart.
	–min-arc-time=ARG	The minimum length of time (in seconds) that a sequence of observations must span to be considered as an arc. The default value is 60.0 seconds.
	–min-arc-gap=ARG	The minimum length of time (in seconds) between two arcs for them to be considered separate arcs. The default value is 60.0 seconds.
	–min-arc-length=ARG	The minimum number of epochs that can be considered an arc. The default value is 5 epochs.
	–noise=ARG	The noise threshold used in finding discontinuitites. The default is 0.1000 cycles.
-b	–elev-bin=ARG	Range of elevations to use in computing the statistical summaries. Repeat to specify multiple bins. The default is "-b 0-10 -b 10-20 -b 20-60 -b 10-90".
-c	–msc=ARG	Station coordinate file.
-p	–pos=ARG	Location of the antenna in meters ECEF.
-E	–health-src=ARG	Do not use data from unhealthy SVs as determined using this ephemeris source. Can be RINEX navigation or FIC file(s).
	–strip=ARG	Factor used in stripping data prior to computing descriptive statistics. The default value is 3.2.

	–phase=ARG	Only compute phase double differences.
-S	–SNR=ARG	Only include observables with a raw signal strength, or SNR, of at least this value, in dB. The default is 20 dB.
-m	–msid=NUM	Station to process data for. Used to select a station position from the msc file or data from a SMODF file.
-w	–window=NUM	Compute mean values of the double differences over this time span (seconds). (15 min = 900)
-r	–raw	Output the raw double differences in addition to the descriptive statistics.
-a	–all-combos	Compute all combinations, don't just use one master SV.
-n	–near	Allow the program to select an ephemeris that is not strictly in the future. Only affects the selection of which broadcast ephemeris to use. i.e. use a close ephemeris.
	–zero-trop	Disables trop corrections.

Examples

```
> ddGen -1 obs/s121001a.09o -2 obs/s121001a.09o -e nav/s121001a.09n

>s  ObsID                elev     noise(mad)    median     # DDE     # SVE    kurt   jumps
>s -----------------     -----    ----------  ----------  -------   -------  ------  -----
>s L1 GPSC/A pseudorange  0-10    0.0000000   0.000e+00     4476      4476    -nan    0
>s L1 GPSP pseudorange    0-10    0.0000000   0.000e+00     4289      4476    -nan    0
>s L1 GPSP phase          0-10    0.0000000   0.000e+00     4289      4476    -nan    0
>s L1 GPSP doppler        0-10    0.0000000   0.000e+00     4289      4476    -nan    0

>s L1 GPSC/A pseudorange 10-20    0.0000000   0.000e+00     5603      5752    -nan    0
>s L1 GPSP pseudorange   10-20    0.0000000   0.000e+00     5584      5752    -nan    0
>s L1 GPSP phase         10-20    0.0000000   0.000e+00     5584      5752    -nan    0
>s L1 GPSP doppler       10-20    0.0000000   0.000e+00     5584      5752    -nan    0

>s L1 GPSC/A pseudorange 20-60    0.0000000   0.000e+00    14499     16507    -nan    0
>s L1 GPSP pseudorange   20-60    0.0000000   0.000e+00    14496     16507    -nan    0
>s L1 GPSP phase         20-60    0.0000000   0.000e+00    14496     16507    -nan    0
>s L1 GPSP doppler       20-60    0.0000000   0.000e+00    14496     16507    -nan    0
```

3.33.6 Data Input

Several different types of data are required to compute these residuals; the raw observations, the receiver antenna position, the satellite position, and optionally weather observations. The raw observations may be supplied to reszilla in one of several formats; rinex obs (see RinexObsData class), smodf (see SMODFData class), and MDP (see MDPObsEpoch class in apps/MDPtools). The reciever antenna postion may be specified in the rinex obs header or via a station coordinates file (see MSCData class).

3.33.7 Output

There are two general types of output that reszilla produces - statistical summaries and the raw residuals. The mean, standard deviation, and maximum value of the residuals are calculated as a function of specified elevation ranges and are output in a statistics table. Looking at the results for each elevation bin is useful as ORDs tend to be much higher when satellites are lower on the horizon. For a more thorough analysis, the ORD or DD residuals calculated by reszilla may be output in a matrix format to a file with columns for time, PRN, elevation, ORD or clock residual, IODC, satellite health, and a flag for the residual type. The flag specifies exactly which of the 13 possible residual types the data on that row represent, depending on the method used for calculation.

One benefit of this output feature is that residuals can be looked at for particular time periods or PRNs. Fortunately there is a companion plotting tool that makes this simple. Given a reszilla output file, the dplot program will plot residuals and, if specified, receiver clock estimates versus time using gnuplot. A user may specify the time range, stripping value, and PRN(s) to use in the plot, as well as a filename for saving the result.

```
Types in the raw output files:
   0 - c1p2 observed range deviation
   50 - computed clock, difference from estimate, strip
   51 - linear clock estimate, abdev
Double difference types:
   10 - c1     20 - c2
   11 - p1     21 - p2
   12 - l1     22 - l2
   13 - d1     23 - d2
   14 - s1     24 - s2
```

3.33.8 Notes

The criteria min-arc-time and min-arc-length are both required to be met for an arc to be valid in double difference mode. All output quantities (stddev, min, max, ord, clock, double difference, ...) are in meters.

3.34 *rmwcheck rnwcheck rowcheck*

3.34.1 Overview

These applications read a RINEX observation (*rowcheck*), navigation(*rnwcheck*), or meteorological (*rmwcheck*) data file and check it for errors.

3.34.2 Usage

rmwcheck rnwcheck rowcheck

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-l	–quit-on-first-error	Quit on the first error encountered.
-t	–time=TIME	Time of first record to count (Default = BOT).
-e	–end-time=TIME	End of time range to compare (Default = EOT).

rmwcheck usage: rmwcheck [options] <RINEX Met file>
rnwcheck usage: rnwcheck [options] <RINEX Nav file>
rowcheck usage: rowcheck [options] <RINEX Obs file>

3.34.3 Examples

```
> rnwcheck -t "08/01/2006 12:00:00" -e "08/01/2006 15:00:00" s081214a.99n

Checking s081213a.99n
Read 200 records.
```

3.34.4 Notes

Only the first error in each file is reported. The entire file is always checked regardless of time options.

3.35 *rmwdiff rnwdiff rowdiff*

3.35.1 Overview

These applications difference RINEX observation, navigation, and meteorological data files.

3.35.2 Usage

rmwdiff rnwdiff rowdiff

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-l	–quit-on-first-error	Quit on the first error encountered.
-t	–time=TIME	Start of time range to compare (Default = BOT.)
-e	–end-time=TIME	End of time range to compare (Default = EOT.)

rmwdiff usage: rmwdiff [options] <RINEX Met file> <RINEX Met file>
rnwdiff usage: rnwdiff [options] <RINEX Nav file> <RINEX Nav file>
rowdiff usage: rowdiff [options] <RINEX Obs file> <RINEX Obs file>

3.35.3 Examples

```
> rowdiff obs/s121001a.01o obs/s121001a.02o

Comparing the following fields (other header data is ignored):
C1 D1 D2 L1 L2 P1 P2
<Dump of RinexObsData - time:  01  1  1  0  0  0.0000000 epochFlag:  0 numSvs: 11 clk offset:
0.000000
Sat G01 C1: 21623650.706/0/8 D1:    -1740.071/0/8 D2:    -1355.897/0/8 L1: -17390026.255/0/8
L2: -13535827.656/0/8 P1: 21623650.392/0/8 P2: 21623657.569/0/8
Sat G03 C1: 20805015.215/0/8 D1:    -1654.577/0/8 D2:    -1289.282/0/8 L1: -22641755.914/0/8
L2: -17618096.770/0/8 P1: 20805015.003/0/8 P2: 20805021.105/0/8
Sat G11 C1: 24129742.024/0/7 D1:     3245.246/0/7 D2:     2528.744/0/7 L1: -4672870.369/0/7
L2: -3626228.611/0/7 P1: 24129741.782/0/7 P2: 24129750.888/0/7
Sat G13 C1: 22087276.186/0/8 D1:        7.400/0/8 D2:        5.765/0/8 L1: -16451815.112/0/8
L2: -12553265.286/0/8 P1: 22087276.610/0/8 P2: 22087282.441/0/8
Sat G15 C1: 23463116.796/0/7 D1:     -497.311/0/8 D2:     -387.518/0/8 L1: -9031186.781/0/8
L2: -7031551.474/0/8 P1: 23463116.213/0/8 P2: 23463124.003/0/8
Sat G19 C1: 21324621.372/0/8 D1:     2187.448/0/8 D2:     1704.503/0/8 L1: -18645307.237/0/8
L2: -14518504.343/0/8 P1: 21324621.390/0/8 P2: 21324628.098/0/8
Sat G22 C1: 22350863.766/0/7 D1:    -1204.472/0/8 D2:     -938.550/0/8 L1: -12632952.524/0/8
L2: -9804132.252/0/8 P1: 22350863.282/0/8 P2: 22350870.038/0/8
Sat G25 C1: 24578217.445/0/7 D1:    -3164.811/0/7 D2:    -2466.069/0/7 L1: -3829204.504/0/7
L2: -2958619.116/0/7 P1: 24578217.563/0/7 P2: 24578226.318/0/7
Sat G27 C1: 23262592.158/0/7 D1:     2951.056/0/8 D2:     2299.519/0/8 L1: -9166691.680/0/8
```

```
L2: -7120447.504/0/8 P1: 23262592.029/0/8 P2: 23262598.552/0/8
Sat G28 C1: 21283503.220/0/8 D1:     -585.103/0/8 D2:     -455.924/0/8 L1: -17698942.286/0/8
L2: -13775959.458/0/8 P1: 21283503.017/0/8 P2: 21283507.983/0/8
Sat G31 C1: 20803601.031/0/8 D1:      878.855/0/8 D2:      684.823/0/8 L1: -22576510.085/0/8
L2: -17577293.102/0/8 P1: 20803600.689/0/8 P2: 20803606.968/0/8

...

> rnwdiff nav/s121001a.01n nav/s121001a.02n
<PRN:  1 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  136
HOWtime:  86406
<PRN:  3 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  186
HOWtime:  86406
<PRN: 11 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   18
HOWtime:  86406
<PRN: 13 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   28
HOWtime:  86406
<PRN: 15 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  226
HOWtime:  86406
<PRN: 19 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  195
HOWtime:  86406
<PRN: 22 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  135
HOWtime:  86406
<PRN: 25 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   29
HOWtime:  86406
<PRN: 27 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   66
HOWtime:  86406
<PRN: 28 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   28
HOWtime:  86406
<PRN: 31 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   91
HOWtime:  86406
<PRN:  8 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  149
HOWtime:  88716

...

>  rmwdiff met/412_001a.00m met/412_001a.01m
Comparing the following fields (other header data is ignored):
PR TD HR
<  2451545 00000000 0.000000000000000 Any
  PR 860.3
  TD 17.2
  HR 95.5
<  2451545 00900000 0.000000000000000 Any
  PR 860.1
  TD 17.2
```

```
  HR 95.8
<  2451545 01800000 0.000000000000000 Any
  PR 859.9
  TD 17.2
  HR 96
<  2451545 02700000 0.000000000000000 Any
  PR 859.6
  TD 17.1
  HR 96.2
```

3.35.4 Notes

Only the first error in each file is reported. The entire file is always checked regardless of time options.

3.36 *RinDump*

3.36.1 Overview

The application reads a RINEX file and dumps the obervation types in columns. Output is to the screen, with one time tag and one satellite per line.

3.36.2 Usage

RinDump

Optional Arguments

Short Arg.	Long Arg.	Description
	–pos	Output only positions from aux headers; sat and obs are ignored.
-n	–num	Make output purely numeric (no header, no system char on sats).
	–format <file>	Output times in CommonTime format (Default: %4F %10.3g).
	–file <file>	RINEX observation file; this option may be repeated.
	–obs <obs>	RINEX observation type, found in file header.
	–sat <sat>	RINEX satellite ID (e.g. G31 for GPS PRN 31).
-h	–help	Print this and quit.

```
RinDump usage: RinDump [-n] <rinex obs file> [<satellite(s)> <obstype(s)>]

The optional argument -n tells RinDump its output should be purely numeric.
```

3.36.3 Examples

```
> RinDump algo1580.06o 3 4 5

# Rinexdump file: algo1580.06o Satellites: G03 G04 G05 Observations: ALL
# Week GPS_sow Sat          L1 L S          L2 L S          C1 L S
1378 259200.000 G03 -3843024.647 0 3 -2994560.443 0 1 23796436.087 0 0
1378 259230.000 G03 -3954052.735 0 3 -3081075.654 0 2 23775308.750 0 0
1378 259260.000 G03 -4064994.465 0 2 -3167523.561 0 3 23754197.617 0 0

 . . .

        P2 L S          P1 L S          S1 L S          S2 L S
23796439.457 0 0 23796436.350 0 0     21.100 0 0      11.000 0 0
23775311.168 0 0 23775308.182 0 0     22.100 0 0      17.800 0 0
23754199.648 0 0 23754196.550 0 0     17.000 0 0      18.600 0 0

 . . .
```

3.36.4 Notes

MATLAB and Octave can read the purely numeric output.

3.37 *RinNav*

3.37.1 Overview

This application reads one or more RINEX (v.2+) navigation files and writes the merged navigation data to one or more output (ver 2 or 3) files. A summary of the ephemeris data may be written to the screen.

3.37.2 Usage

RinNav

```
RinNav usage: RinNav [options] <file>
```

Required Arguments

Short Arg.	Long Arg.	Description
	–file <fn>	Name of file with more options [#->EOL = comment] [repeat]
	–nav <file>	Input RINEX navigation file name [repeat]
	–navpath <p>	Path of input RINEX navigation file(s)

Optional Arguments

Short Arg.	Long Arg.	Description
	–start <t[:f]>	Start processing data at this epoch ([Beginning of dataset])
	–stop <t[:f]>	Stop processing data at this epoch ([End of dataset])
	–exSat <sat>	Exclude satellite [system] from output [e.g. G17,R] [repeat]
	–out <[sys,]fn>	Output [system sys only] to RINEX ver. 3 file fn
	–out2 <[sys,]fn>	Version 2 output [system sys only] to RINEX file fn
	–timefmt <fmt>	Format for time tags (see GPSTK::Epoch::printf) in output (%4F %10.3g)
	–ver2	Write out RINEX version 2
	–verbose	Print extra output information
	–debug	Print debug output at level 0 [debug¡n¿ for level n=1-7]
-h	–help	Print this and quit.

3.37.3 Examples

```
 RinNav nav/s121001a.02n nav/s121001a.03n
# RinNav, part of the GPS Toolkit, Ver 1.1 2/2/12, Run 2012/07/23 14:36:05
Dump Rinex3EphemerisStore:
Dump of FileStore
 File  1: nav/s121001a.02n (header for this file follows)
---------------------------------- REQUIRED ----------------------------------
```

```
Rinex Version  2.10,  File type NAVIGATION, System G: (GPS).
Prgm: RinexNavWriter,  Run: 1-02-02  0:05:09,  By: NIMA
(This header is VALID RINEX version 2).
---------------------------------- OPTIONAL ----------------------------------
 Leap seconds is NOT valid
-------------------------------- END OF HEADER -------------------------------
 File  2: nav/s121001a.03n (header for this file follows)
---------------------------------- REQUIRED ----------------------------------
Rinex Version  2.10,  File type NAVIGATION, System G: (GPS).
Prgm: RinexNavWriter,  Run: 1-02-03  0:05:09,  By: NIMA
(This header is VALID RINEX version 2).
---------------------------------- OPTIONAL ----------------------------------
 Leap seconds is NOT valid
-------------------------------- END OF HEADER -------------------------------
End dump of FileStore
Dump of GPSEphemerisStore:
  BCE map for satellite 1 has 16 entries.
PRN  1 TOE 1147 180000.000 GPS TOC 180000.000 HOW 172806.000 KEY 1147 172800.000 GPS
PRN  1 TOE 1147 194400.000 GPS TOC 194400.000 HOW 194346.000 KEY 1147 187200.000 GPS
PRN  1 TOE 1147 201600.000 GPS TOC 201600.000 HOW 194406.000 KEY 1147 194400.000 GPS
PRN  1 TOE 1147 208800.000 GPS TOC 208800.000 HOW 201606.000 KEY 1147 201600.000 GPS
PRN  1 TOE 1147 244800.000 GPS TOC 244800.000 HOW 244086.000 KEY 1147 237600.000 GPS
PRN  1 TOE 1147 251984.000 GPS TOC 251984.000 HOW 246216.000 KEY 1147 244784.000 GPS
PRN  1 TOE 1147 252000.000 GPS TOC 252000.000 HOW 244806.000 KEY 1147 244800.000 GPS
PRN  1 TOE 1147 259184.000 GPS TOC 259184.000 HOW 259146.000 KEY 1147 251984.000 GPS
PRN  1 TOE 1199 266384.000 GPS TOC 266384.000 HOW 259206.000 KEY 1199 259184.000 GPS
PRN  1 TOE 1199 273600.000 GPS TOC 273600.000 HOW 271296.000 KEY 1199 266400.000 GPS
PRN  1 TOE 1199 280800.000 GPS TOC 280800.000 HOW 280746.000 KEY 1199 273600.000 GPS
```

3.38 *RinEdit*

3.38.1 Overview

The application opens and reads RINEX observation files(s) (v2+), applies editing commands, and write out the modified RINEX data to RINEX v3 file(s).

3.38.2 Usage

RinEdit

Optional Arguments

Short Arg.	Long Arg.	Description
	–IF <f>	Input RINEX observation file names [repeat]
	–ID <p>	Path of input RINEX observation file(s)
	–OF <fn>	Output RINEX obs files [also see –OF <f,t> below]
	–OD <p>	Path of output RINEX observation file(s)
	–file <fn>	Name of file containing more options [#->EOL = comment]
	–log <fn>	Output log file name
	–ver2	Write out RINEX version 2
	–verbose	Print extra output information
	–debug	Print debug output at level 0 [debug<n> for level n=1-7]
	–help	Print syntax and editing command page

3.38.3 Examples

```
> RinEdit --IF acor1480.08o --IF areq015o.10o --OF out.12o --verbose

# RinEdit, part of the GPS Toolkit, Ver 1.0 8/1/11, Run 2012/07/09 12:17:20
 Edit cmd: OF_Output_File 0 SV:?-1 OT: d:0.0000 i:0 t:BeginTime >out.12o<
Reading header...
Reading observations...
 Opened output file out.12o at time 2008/05/27 00:00:00 = 1481 172800.000 GPS
Reading header...
Reading observations...
```

3.39 *rinexpvt*

3.39.1 Overview

The application generates a user position based on RINEX observation data with the option of including navigation and meteorological data to aid error correction.

3.39.2 Usage

rinexpvt

Required Arguments

Short Arg.	Long Arg.	Description
-o	–obs-file=ARG	RINEX observation file.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-n	–nav-file=ARG	RINEX navigation file. Required for single frequency ionosphere correction.
-p	–pe-file=ARG	SP3 Precise Ephemeris file. Repeat this for each input file.
-m	–met-file=ARG	RINEX meteorological file.
-t	–time-format=ARG	Alternate time format string.
-e	–enu=ARG	Use the following as origin to solve for East/North/Up coordinates, formatted as a string: “X Y Z”.
-l	–elevation-mask=ARG	Elevation mask (degrees).
-g	–logfile=ARG	Write logfile to this file.
-r	–rate=ARG	Observation interval (Default = 30 seconds or Rinex Header specification).
-y	-yuma=ARG	Yuma almanac file.
-a	–sem=ARG	SEM almanac file.
-s	–single-frequency	Use only C1 (SPS).
-f	–dual-frequency	Use only P1 and P2 (PPS).
-i	–no-ionosphere	Do NOT correct for ionosphere delay.
-x	–no-closest-ephemeris	Allow ephemeris use outside of fit interval.
-c	–no-carrier-smoothing	Do NOT use carrier phase smoothing.
-z	–no-glonass	Exclude GLONASS Satellites from PVT solution.

3.39.3 Examples

```
> rinexpvt -o arl2800.06o -n arl2800.06n
2006 1 1 09 41 00 918130.968492 -4346073.94224 4561982.02123 333.303358692
2006 1 1 09 41 30 918130.956684 -4346073.91529 4561982.01659 333.317002144
2006 1 1 09 42 00 918130.924146 -4346073.83279 4561982.01338 333.279239604
```

3.39.4 Notes

Though not stated in the required options lists, either a RINEX navigation file or an SP3 Precise Ephemeris File is needed, using the -n or -p option respectively. When using precise ephemeris, three files must be included: the previous day, the current day, and the next day.

Although -z argument appears as optional, in this release, it is always turned on, but implementation will occur in a later release.

3.40 *RinSum*

3.40.1 Overview

The application reads a RINEX file and summarizes its content.

3.40.2 Usage

RinSum

Optional Arguments

Short Arg.	Long Arg.	Description
-i	–input	Input file name(s).
-f		File containing more options.
-o	–output	Output file name.
-p	–path	Path for input file(s).
-R	–Replace	Replace header with full one.
-s	–sort	Sort the PRN/Obs table on begin time.
-g	–gps	Print times in the PRN/Obs table as GPS times.
	–gaps	Print a table of gaps in the data, assuming specified interval dt.
	–start	Start time: <time> is 'GPSweek,sow' OR 'YYYY,MM,DD,HH,Min,Sec'.
	–stop	Stop time: <time> is 'GPSweek,sow' OR 'YYYY,MM,DD,HH,Min,Sec'.
-b	–brief	Produce a brief (6-line) summary.
-h	–help	Print syntax and quit.
-d	–debug	Print debugging information.

3.40.3 Examples

```
>RinSum obs/s051001a.04o
# RinSum, part of the GPS Toolkit, Ver 3.3 1/31/12, Run 2012/07/17 11:12:32
+++++++++++++ RinSum summary of Rinex obs file obs/s051001a.04o +++++++++++++
-------------------------------- REQUIRED ---------------------------------
Rinex Version 2.10, File type Observation, System G (GPS).
Prgm: GFW - ROW, Run: 12/31/2003 23:59:53, By: NIMA
Marker name: 85405, Marker type: .
Observer : Monitor Station, Agency: NIMA
Rec#: 1, Type: ZY12, Vers:
Antenna # : 85405, Type : AshTech Geodetic 3
Position     (XYZ,m) : (3633910.6680, 4425277.7563, 2799862.8708).
Antenna Delta (HEN,m) : (0.0000, 0.0000, 0.0000).
Wavelength factor L1: 1 L2: 1
GPS Observation types (9):
 Type #01 (L1P) L1 GPSP phase
 Type #02 (L2P) L2 GPSP phase
 Type #03 (C1C) L1 GPSC/A pseudorange
 Type #04 (C1P) L1 GPSP pseudorange
 Type #05 (C2P) L2 GPSP pseudorange
 Type #06 (D1P) L1 GPSP doppler
 Type #07 (D2P) L2 GPSP doppler
 Type #08 (S1P) L1 GPSP snr
```

```
 Type #09 (S2P) L2 GPSP snr
Time of first obs 2004/01/01 00:00:00.000 Unknown
(This header is VALID)
--------------------------------- OPTIONAL ---------------------------------
Marker number : 85405
Signal Strenth Unit =
Comments (1) :
Data are thinned (not smoothed) 30s. observations
------------------------------- END OF HEADER -------------------------------

Reading the observation data...
Computed interval 30.00 seconds.
Computed first epoch: 2004/01/01 00:00:00 = 1251 4 345600.000
Computed last epoch: 2004/01/01 23:59:30 = 1251 4 431970.000
Computed time span: 23h 59m 30s = 86370 seconds.
Computed file size: 3785956 bytes.
There were 2880 epochs (100.00% of 2880 possible epochs in this timespan) and 0 inline header blocks.

     Summary of data available in this file: (Spans are based on times and interval)
System G = GPS:
 Sat\OT: L1P   L2P   C1C   C1P   C2P   D1P   D2P   S1P   S2P  Span           Begin time - End time
 G01    945   945   942   945   945   945   945   945   945  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G02    911   906   889   911   906   911   906   911   906  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G03    872   869   855   872   869   872   869   872   869  2433  2004/01/01 00:42:00 - 2004/01/01 20:58:00
 G04    914   908   884   914   908   914   908   914   908  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G05    785   785   781   785   785   785   785   785   785   860  2004/01/01 07:19:30 - 2004/01/01 14:29:00
 G06    890   890   885   890   890   890   890   890   890   947  2004/01/01 09:31:00 - 2004/01/01 17:24:00
 G07    735   735   735   735   735   735   735   735   735   735  2004/01/01 03:51:00 - 2004/01/01 09:58:00
 G08    924   923   916   924   923   924   923   924   923   974  2004/01/01 00:17:00 - 2004/01/01 08:23:30
 G09    665   665   659   665   665   665   665   665   665  1310  2004/01/01 06:34:30 - 2004/01/01 17:29:00
 G10    947   943   937   947   943   947   943   947   943  1407  2004/01/01 02:44:00 - 2004/01/01 14:27:00
 G11    699   696   657   699   696   699   696   699   696  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G13    890   888   875   890   888   890   888   890   888  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G14    700   700   698   700   700   700   700   700   699   702  2004/01/01 16:14:00 - 2004/01/01 22:04:30
 G15    903   903   898   903   903   903   903   903   903   959  2004/01/01 13:08:00 - 2004/01/01 21:07:00
 G16    982   982   969   982   982   982   982   982   982  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G17    738   738   737   738   738   738   738   738   738   766  2004/01/01 08:20:30 - 2004/01/01 14:43:00
 G18    773   772   765   773   772   773   772   773   772   873  2004/01/01 12:43:30 - 2004/01/01 19:59:30
 G20    789   789   771   789   789   789   789   789   789  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G21    930   930   921   930   930   930   930   930   930   966  2004/01/01 11:02:00 - 2004/01/01 19:04:30
 G22    793   793   793   793   793   793   793   793   793   793  2004/01/01 15:15:30 - 2004/01/01 21:51:30
 G23    877   877   888   877   877   877   877   877   877   904  2004/01/01 14:52:00 - 2004/01/01 22:23:30
 G24    675   675   667   675   675   675   675   675   675  1394  2004/01/01 01:29:30 - 2004/01/01 13:06:00
 G25    652   651   647   652   651   652   651   652   651  1412  2004/01/01 12:07:30 - 2004/01/01 23:53:00
 G26    927   927   920   927   927   927   927   927   927  1411  2004/01/01 04:47:30 - 2004/01/01 16:32:30
 G27    954   954   951   954   954   954   954   954   954  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G28    882   881   876   882   881   882   881   882   881   914  2004/01/01 01:41:30 - 2004/01/01 09:18:00
 G29    885   884   870   885   884   885   884   885   884  1419  2004/01/01 03:57:30 - 2004/01/01 15:46:30
 G30    701   701   700   701   701   701   701   701   701   749  2004/01/01 09:27:30 - 2004/01/01 15:41:30
 G31    973   970   962   973   970   973   970   973   970  2541  2004/01/01 00:32:00 - 2004/01/01 21:42:00
TOTAL  24311 24280 24048 24311 24280 24311 24280 24311 24279
```

3.41 *rtAshtech*

3.41.1 Overview

This application logs observations from an Ashtech Z-XII receiver. It records observations directly into the RINEX format. A number of optional outputs are possible. The raw messages from a receiver can be recorded. Observations can also be recorded in a format that is easily imported into numerical packages.

3.41.2 Usage

Optional Arguments

Short Arg.	Long Arg.	Description
-h	–help	Print help usage.
-v	–verbose	Increased diagnostic messages.
-r	–raw	Record raw observations.
-l	–log	Record log entries.
-t	–text	Record observations as simple text files.
-o	–rinex-obs=ARG	Naming convention for RINEX obs files.
-n	–rinex-nav=ARG	Naming convention for RINEX nav message files.
-T	–text-obs=ARG	Naming convention for obs in simple text files.
-i	–input	Where to read ashTech data. Can be a file or a serial device (ser:/dev/ttyS0), a tcp port (tcp:hostname:port), or standard input (the default).

3.41.3 Examples

```
> rtAshtech -p /dev/ttyS1
```

```
> rtAshtech -o "minute\%03j\%02H\%02M.\%02yo"
```

3.41.4 Notes

rtAshtech only works on UNIX systems with POSIX compliant serial ports.

3.42 *sp3version*

3.42.1 Overview

This application reads an SP3 file (either a or c format) and writes it to another file (also either in a or c format).

3.42.2 Usage

sp3version

Optional Arguments

Short Arg.	Long Arg.	Description
	–in	A file from which to take the input. The default is stdin.
	–out	A file into which to write the output. The default is sp3.out.
	–outputC	Output version c (otherwise a).
	–msg	Add message as a comment to the output header.
	–verbose	Output to screen: dump headers, data, etc.

3.42.3 Examples

```
  sp3version --in sp3/igs13355.sp3 --verbose
Reading file sp3/igs13355.sp3
Input SP3 Header: version SP3a containing positions only.
 Time tag : 2005/08/12  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
 Data used as input : ORBIT
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 List of satellite PRN/accuracy (29 total) :
 G01/3 G02/4 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3
 G09/3 G10/3 G11/3 G13/3 G14/3 G15/3 G16/0 G18/3
 G19/3 G20/3 G21/3 G22/3 G23/3 G24/3 G25/3 G26/3
 G27/3 G28/3 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header

Output SP3 Header: version SP3c containing positions only.
 Time tag : 2005/08/12  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
```

```
 Data used as input : ORBIT
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 File type: 'G' which is GPS
 Time System: GPS
 Base for Pos/Vel = 1.2500000
 Base for Clk/Rate = 1.025000000
 List of satellite PRN/accuracy (29 total) :
 G01/3 G02/4 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3
 G09/3 G10/3 G11/3 G13/3 G14/3 G15/3 G16/0 G18/3
 G19/3 G20/3 G21/3 G22/3 G23/3 G24/3 G25/3 G26/3
 G27/3 G28/3 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header

Input:
* G-1 2005/08/12  0:00:00.000 = 1335/432000.000
Output sdev 1975 3351 7681 2777
Output correl 55396995 47739704 62887091 36478446 51340090 95222971
Output:
* G-1 2005/08/12  0:00:00.000 = 1335/432000.000
Input:
P G01 2005/08/12  0:00:00.000 = 1335/432000.000 X=  15202.734861 Y=   1913.732043 Z= -21514.72055
C=      2.747556 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
Output sdev 1566 4009 1297 1087
Output correl 99892450 21825690 51293238 83911222 61263982 29603161
Output:
P G01 2005/08/12  0:00:00.000 = 1335/432000.000 X=  15202.734861 Y=   1913.732043 Z= -21514.72055
C=      2.747556 sX=90 sY=62 sZ=71 sC= 14 clockEvent - - -
    and EP cXX=1566 cYY=4009 cZZ=1297 cCC=   1087 cXY=99892450 cXZ=21825690 cXC=51293238 cYZ=8391
cYC=61263982 cZC=29603161
Input:
P G02 2005/08/12  0:00:00.000 = 1335/432000.000 X= -21564.807909 Y=  10266.659247 Z= -11746.80534
C=    -27.138828 sX= 0 sY= 0 sZ= 0 sC=  0 clockEvent - - -
Output:
P G02 2005/08/12  0:00:00.000 = 1335/432000.000 X= -21564.807909 Y=  10266.659247 Z= -11746.80534
C=    -27.138828 sX=63 sY=51 sZ=48 sC= 96 - clockPrediction orbitManeuver orbitPrediction
Input:
P G03 2005/08/12  0:00:00.000 = 1335/432000.000 X=  14716.409703 Y=  -5992.688052 Z=  21147.41312
C=     25.193262 sX= 0 sY= 0 sZ= 0 sC=  0 - clockPrediction orbitManeuver orbitPrediction
Output:
```

3.43 *svvis*

3.43.1 Overview

This application computes when satellites are visible at a given point on the earth.

3.43.2 Usage

svvis

Required Arguments

Short Arg.	Long Arg.	Description
-e	–eph=ARG	Where to get the ephemeris data. Can be RINEX, nav, FIC, MDP, SP3, YUMA, and SEM.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
	–elevation-mask=ARG	The elevation above which an SV is visible. The default is 0 degrees.
-p	–position=ARG	Receiver antenna position in ECEF (x,y,z) coordinates. Format as string: “X Y Z”.
-c	–msc=ARG	Station coordinate file.
-m	–msid=ARG	Station number to use from the msc file.
	–graph-elev=ARG	Output data at the specified interval. Interval is in seconds.
-l	–time-span=ARG	How much data to process, in seconds. Default is 86400.
	–start-time=TIME	When to start computing positions. The default is the start of the ephemeris data.
	–stop-time=TIME	When to stop computing positions. The default is one day after the start time.
	–print-elev	Print the elevation of the sv at each change in tracking. The default is just to ouput the PRN of the SV.
	–rise-set	Print the visibility data by PRN in rise-set pairs.
	–tabular	Print the visibility data in a tabular format.
	–recent-eph	Use this if the ephemeris data provided uses 10-bit GPS weeks and it should be converted to the current epoch or to the epoch current to the “start-time”, if specified.

3.43.3 Examples

```
> svvis -e nav/s121001a.09n -p -3939182.6018,3467075.4175,-3613220.2782,402 --tabular

SEARCH_INTERVAL: 2009 001 00:00:00 to 2009 002 02:00:00
ELEVATION_CUTOFF: 0.000
#     Rise (Yr DOY HMS) Set  (Yr DOY HMS) El Sys          Parameters
PASS: 2009 001 00:12:55 2009 001 03:59:59 80 GPS          PRN=23
```

```
PASS: 2009 001 01:23:30 2009 001 05:59:43 62 GPS          PRN=25
PASS: 2009 001 01:37:18 2009 001 05:59:59 77 GPS          PRN=13
PASS: 2009 001 02:33:15 2009 001 03:59:59 16 GPS          PRN=28
PASS: 2009 001 02:41:00 2009 001 05:59:59 53 GPS          PRN=07
PASS: 2009 001 03:48:56 2009 001 05:59:59 43 GPS          PRN=17
PASS: 2009 001 04:14:43 2009 001 07:59:59 77 GPS          PRN=08
PASS: 2009 001 04:41:56 2009 001 07:59:59 78 GPS          PRN=27
PASS: 2009 001 05:26:57 2009 001 07:59:59 56 GPS          PRN=04
PASS: 2009 001 07:00:06 2009 001 11:59:59 25 GPS          PRN=26
PASS: 2009 001 07:16:58 2009 001 07:59:59 15 GPS          PRN=02
...
```

3.44 *TECMaps*

3.44.1 Overview

Program TECMaps reads RINEX data files containing extended RINEX observation types EL, AZ and SR or VR from several sites and at each epoch fits the vertical TEC data to a model of the ionosphere on a two-dimensional grid surface. Hardware TEC measurement biases are corrected, using input from the program IonoBias. The user can specify the type of grid, the type of TEC data and the model to be used. Output is in the form of files, one per epoch, which can be used to plot the 2D ionospheric TEC surface.

3.44.2 Usage

TECMaps

Required Arguments

Short Arg.	Long Arg.	Description
	–input	Input RINEX obs file name(s).

Optional Arguments

Short Arg.	Long Arg.	Description
-f		File containing more options.

Reference Station Position (One Required)

Short Arg.	Long Arg.	Description
	–RxLLH <l,l,h>	Reference site position in geodetic lat, lon (E), ht (deg,deg,m).
	–RxXYZ <x,y,z>	Reference site position in ECEF coordinates (m).
	–inputdir	Path for input file(s).

Ephemeris Input

Short Arg.	Long Arg.	Description
	–navdir	Path of navigation file(s).
	–nav	Navigation (RINEX navigation OR SP3) file(s).

Output

Short Arg.	Long Arg.	Description
	–log	Output log file name.

Time Limits

Short Arg.	Long Arg.	Description
	–BeginTime	Start time, arg is of the form YYYY,MM,DD,HH,Min,Sec.
	–BeginGPSTime	Start time, arg is of the form GPSweek,GPSsow.
	–EndTime	End time, arg is of the form YYYY,MM,DD,HH,Min,Sec.
	–EndGPSTime	End time, arg is of the form GPSweek,GPSsow.

Processing

Short Arg.	Long Arg.	Description
	–noVTECmap	Do NOT create the VTEC map.
	–MUFmap	Create MUF map as well as VTEC map.
	–F0F2map	Create F0F2 map as well as VTEC map.
	–Title1 <title>	Title information.

	–Title2 <title>	Second title information.
	–BaseName <name>	Base name for output files.
	–DecorrError <de>	Decorrelation error rate in TECU/1000km (3).
	–Biases <file>	File containing estimated sat+rx biases (Prgm IonoBias).
	–ElevThresh <ele>	Minimum elevation (6 degrees).
	–MinAcqTime<t>	Minimum acquisition time (0 seconds).
	–FlatFit	Flat fit type (default).
	–LinearFit	Linear fit type.
	–IonoHeight <n>	Ionosphere height (km).
	–Offset <tec>	Overall bias to add to data (TECU).

Grid

Short Arg.	Long Arg.	Description
	–UniformSpacing	Grid uniform in space (XYZ) (default).
	–UniformGrid	Grid uniform in Lat and Lon.
	–OutputGrid	Output the grid to file <basename.LL>.
	–GnuplotOutput	Write the grid file for gnuplot (default: for Matlab).
	–NumLat <n>	Number of latitude grid points (40).
	–NumLon <n>	Number of longitude grid points (40).
	–BeginLat <lat>	Beginning latitude (21 degrees).
	–BeginLon <lon>	Beginning longitude (230 degrees E).
	–DeltaLat <del>	Grid spacing in latitude (0.25 degrees).
	–DeltaLon <del>	Grid spacing in longitude (1.0 degrees).

Other Options

Short Arg.	Long Arg.	Description
	–XSat	Exclude this satellite (<sat> may be <system> only).

Help

Short Arg.	Long Arg.	Description
-v	–verbose	Print extended output info.
-d	–debug	Increase debug level.
-h	–help	Print syntax and summary of input, then quit.

3.44.3 Examples

```
> TECMaps --input obs/s121001a.09o --RxXYZ 3,3,3 --navdir ./nav --nav s121001a.07n

TECMaps, built on the GPSTK ToolKit, Ver 1.2 9/21/07, Run 2012/07/24 19:06:19
TECMaps output directed to log file vtm.log
TECMaps timing: 1.850 seconds.

-------------------------------vtm.log file-----------------------------------

TECMaps, built on the GPSTK ToolKit, Ver 1.2 9/21/07, Run 2012/07/24 19:06:19

Input file #1: obs/s121001a.09o

End of loop over stations to read headers.
```

```
Process at time = 2009/1/1 0:0: 0.000=1512/345600.000
0 data at epoch 2009/1/1 0:0: 0.000=1512/345600.000, file #1.
Process at time = 2009/1/1 0:0:30.000=1512/345630.000
0 data at epoch 2009/1/1 0:0:30.000=1512/345630.000, file #2.
Process at time = 2009/1/1 0:1: 0.000=1512/345660.000
0 data at epoch 2009/1/1 0:1: 0.000=1512/345660.000, file #3.
```

3.44.4 Notes

Input is on the command line, or of the same format in a file (-f<file>).

3.45 *timeconvert*

3.45.1 Overview

This application allows the user to convert between time formats associated with GPS. Time formats include: civilian time, Julian day of year and year, GPS week and seconds of week, Z counts, and Modified Julian Date (MJD).

3.45.2 Usage

timeconvert

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-A	–ansi=TIME	“ANSI-Second”.
-c	–civil=TIME	“Month(numeric) DayOfMonth Year Hour:Minute:Second
-R	–rinex-file=TIME	“Year(2-digit) Month(numeric) DayOfMonth Hour Minute Second”.
-o	–ews=TIME	“GPSEpoch 10bitGPSweek SecondOfWeek”.
-f	–ws=TIME	“FullGPSWeek SecondOfWeek”.
-w	–wz=TIME	“FullGPSWeek Zcount”.
	–z29=TIME	“29bitZcount”.
-Z	–z32=TIME	“32bitZcount”.
-j	–julian=TIME	“JulianDate”.
-m	–mjd=TIME	“ModifiedJulianDate”.
-u	–unixtime=TIME	“UnixSeconds UnixMicroseconds”.
-y	–doy=TIME	“Year DayOfYear SecondsOfDay”.
	–input-format=ARG	Time format to use on input.
	–input-time=ARG	Time to be parsed by ”input-format” option.
-F	–format=ARG	Time format to use on output.
-a	–add-offset=NUM	Add NUM seconds to specified time.
-s	–sub-offset=NUM	Subtract NUM seconds from specified time.

3.45.3 Examples

Convert RINEX file time.

```
> timeconvert -R "05 06 1985 13:50:02"
```

```
Month/Day/Year H:M:S            11/06/2010 13:00:00
Modified Julian Date            55506.541666667
GPSweek DayOfWeek SecOfWeek     584 6  565200.000000
FullGPSweek Zcount              1608 376800
Year DayOfYear SecondOfDay      2010 310  46800.000000
Unix: Second Microsecond        1289048400      0
Zcount: 29-bit (32-bit)         306560992 (843431904)
```

Convert ews time.

```
timeconvert -o "01 1379 500"
```

```
Month/Day/Year                  1/25/2026
Hour:Min:Sec                    00:08:20
Modified Julian Date            61065.005787037
GPSweek DayOfWeek SecOfWeek     355 0 500.000000
FullGPSweek Zcount              2403 333
Year DayOfYear SecondOfDay      2026 25 500.000000
Unix_sec Unix_usec              1769299700 0
Zcount: 29-bit (32-bit)         186122573 (1259864397)
```

3.45.4 Notes

If no arguments are given it will convert the current time to all formats. When inputting time values, include quotation marks.

3.46 *vecsol*

3.46.1 Overview

The application computes a 3D vector solution using dual-frequency carrier phases. A double difference algorithm is applied with properly computed weights (elevation sine weighting) and correlations. The program iterates to convergence and attempts to resolve ambiguities to integer values if close enough. Crude outlier rejection is provided based on a triple-difference test. Ephemerides used are either broadcast or precise (SP3).

Alternatively, P code processing is additionally provided. The solution is computed using either the ionosphere-free linear combination, or the average of L1 and L2. The ionospheric model included in broadcast ephemeris may be used. A standard tropospheric correction is applied, or tropospheric parameters (zenith delays) may be estimated for the first station (vector mode) or both.

3.46.2 Usage

vecsol

vecsol usage: vecsol <RINEX Obs file 1> <RINEX Obs file 2>

RINEX Observation Files

The two arguments are names of RINEX observation files. They contain the observations collected at the two end points 1 and 2 of the baseline. They must contain a sufficient set of simultaneous observations to the same satellites.

If no separate station coordinate files are provided, the initial station coordinates are taken from the RINEX headers. Upon finishing, vecsol creates or updates the coordinate file of the first station (vector mode) or both.

Configuration File vecsol.conf

The file vecsol.conf contains the input options for the program, one per line.

Options	Value	Meaning
obsMode	3/2/1/0	If 1 or 3, process carrier phase data (instead of P code data). If 0 or 1, iterate on ionosphere-free vector (not L1 + L2).
truecov	1/0	If 1, use true double difference covariances. If 0, ignore any possible correlations.
precise	1/0	If 1, use precise ephemeris, if 0, use broadcast ephemeris.
iono	1/0	If 1, use the 8-parameter ionospheric model that comes with the broadcast ephemeris (.nav) files.
tropo	1/0	If 1, estimate troposphere parameters (zenith delays relative to the standard value, which is always applied).
vecmode	1/0	If 1, solve the vector, i.e. the three coordinate differences between the baseline end points. If 0, solve for the absolute coordinates of both end points.
debug	1/0	If 1, produce lots of gory debugging output. See the source for what it all means.

refsat elev	number	Minimum elevation (degs) of the reference satellite used for computing inter-satellite differences. Good initial choice: 30.0.
cutoff elev	number	Cut-off elevation (degs). Good initial choice: 10.0 - 20.0.
rej TP, rej TC	two numbers	Phase, code triple differences rejection limit (m).
reduce	1/0	Apply post-reduction to combine dependent unknowns.

Ephemeris File Lists

The file vecsol.nav contains the names of the navigation RINEX files ("nav files", extension). Good navigation RINEX files that are globally valid can be found from the CORS website at http://www.ngs.noaa.gov/CORS/.

The file vecsol.eph contains the names of the precise ephemeris SP3 files (extension .sp3) to be used. These should cover the time span of the observations, with time to spare on both ends. Note that the date in the filenames of the SP3 files is given as GPS week + weekday, not year + day of year, as in the observation and nav files.

In the .nav and .eph files, comment lines have # in the first position.

3.46.3 Examples

```
> vecsol bell0300.02o bell030a.02o

Configuration data from vecsol.conf
-----------------------------------
Use carrier phases:            1
Compute ionosphere-free:       1
Use true correlations:         1
Use precise ephemeris:         1
Use broadcast iono model:      0
Use tropospheric est.:         1
Vector mode:                   1
Debugging mode:                1
Ref sat elevation limit:       30
Cut-off elevation:             20
TD rej. limits (phase, code):  0.1 0.1
Reduce out DD dependencies:    1

Eph file: # skipped
Eph file: igs12851.sp3
Eph file: igs12852.sp3
Eph file: igs12853.sp3
Dump SP3EphemerisStore:
 Reject bad positions.
 Reject bad clocks.
```

```
 Do not reject predicted positions.
 Do not reject predicted clocks.
Dump of FileStore
 File  1: igs12851.sp3 (header for this file follows)
SP3 Header: version SP3a containing positions only.
 Time tag : 2004/08/23  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
 Data used as input : ORBIT
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 List of satellite PRN/accuracy (29 total) :
 G01/3 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3 G09/3
 G10/3 G11/4 G13/4 G14/4 G15/3 G16/3 G17/3 G18/4
 G19/4 G20/4 G21/3 G22/3 G23/4 G24/3 G25/3 G26/3
 G27/3 G28/4 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header
 File  2: igs12852.sp3 (header for this file follows)
SP3 Header: version SP3a containing positions only.
 Time tag : 2004/08/24  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
 Data used as input : ORBIT
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 List of satellite PRN/accuracy (29 total) :
 G01/3 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3 G09/3
 G10/3 G11/4 G13/4 G14/4 G15/3 G16/3 G17/3 G18/3
 G19/4 G20/4 G21/3 G22/3 G23/4 G24/3 G25/3 G26/3
 G27/3 G28/4 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header
 File  3: igs12853.sp3 (header for this file follows)
SP3 Header: version SP3a containing positions only.
 Time tag : 2004/08/25  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
 Data used as input : ORBIT
```

```
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 List of satellite PRN/accuracy (29 total) :
 G01/3 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3 G09/3
 G10/3 G11/4 G13/3 G14/3 G15/3 G16/3 G17/3 G18/3
 G19/3 G20/4 G21/3 G22/3 G23/4 G24/3 G25/3 G26/3
 G27/3 G28/4 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header
End dump of FileStore
Dump of PositionSatStore(1):
 This store does not contain acceleration data.
 Interpolation is Lagrange, of order 10 (5 points on each side)
 Dump of TabularSatStore(1):
  Data stored for 29 satellites
  Time span of data:  FROM 1285 1  86400.000 2004/08/23  0:00:00 Any
      TO 1285 3 344700.000 2004/08/25 23:45:00 Any
  This store contains: position, not velocity, not clock bias, and not clock drift data.
  Checking for data gaps? no
  Checking data interval? no
   Sat GPS 1 : 288 records.
   Sat GPS 3 : 288 records.
   Sat GPS 4 : 288 records.
   Sat GPS 5 : 288 records.
   Sat GPS 6 : 288 records.
   Sat GPS 7 : 288 records.
   Sat GPS 8 : 288 records.
   Sat GPS 9 : 288 records.
   Sat GPS 10 : 276 records.
   Sat GPS 11 : 288 records.
   Sat GPS 13 : 288 records.
   Sat GPS 14 : 288 records.
   Sat GPS 15 : 288 records.
   Sat GPS 16 : 288 records.
   Sat GPS 17 : 288 records.
   Sat GPS 18 : 288 records.
   Sat GPS 19 : 288 records.
   Sat GPS 20 : 288 records.
   Sat GPS 21 : 288 records.
   Sat GPS 22 : 288 records.
   Sat GPS 23 : 288 records.
```

```
    Sat GPS 24 : 288 records.
    Sat GPS 25 : 288 records.
    Sat GPS 26 : 288 records.
    Sat GPS 27 : 288 records.
    Sat GPS 28 : 288 records.
    Sat GPS 29 : 288 records.
    Sat GPS 30 : 288 records.
    Sat GPS 31 : 288 records.
  End dump of TabularSatStore.
End dump of PositionSatStore.
Dump of ClockSatStore(1):
 This store  does not contain clock acceleration data.
 Interpolation is Lagrange, of order 10 (5 points on each side)
 Dump of TabularSatStore(1):
  Data stored for 29 satellites
  Time span of data:  FROM 1285 1  86400.000 2004/08/23  0:00:00 Any
      TO 1285 3 344700.000 2004/08/25 23:45:00 Any
  This store contains: not position, not velocity, clock bias, and not clock drift data.
  Checking for data gaps? no
  Checking data interval? no
   Sat GPS 1 : 288 records.
   Sat GPS 3 : 288 records.
   Sat GPS 4 : 288 records.
   Sat GPS 5 : 288 records.
   Sat GPS 6 : 288 records.
   Sat GPS 7 : 288 records.
   Sat GPS 8 : 288 records.
   Sat GPS 9 : 288 records.
   Sat GPS 10 : 276 records.
   Sat GPS 11 : 288 records.
   Sat GPS 13 : 288 records.
   Sat GPS 14 : 288 records.
   Sat GPS 15 : 288 records.
   Sat GPS 16 : 288 records.
   Sat GPS 17 : 288 records.
   Sat GPS 18 : 288 records.
   Sat GPS 19 : 288 records.
   Sat GPS 20 : 288 records.
   Sat GPS 21 : 288 records.
   Sat GPS 22 : 288 records.
   Sat GPS 23 : 288 records.
   Sat GPS 24 : 288 records.
   Sat GPS 25 : 288 records.
   Sat GPS 26 : 288 records.
   Sat GPS 27 : 288 records.
   Sat GPS 28 : 288 records.
```

```
   Sat GPS 29 : 288 records.
   Sat GPS 30 : 288 records.
   Sat GPS 31 : 288 records.
 End dump of TabularSatStore.
End dump of ClockSatStore.
End dump SP3EphemerisStore.
```

3.46.4 Notes

Currently, vecsol does not recover from cycle slips, so the RINEX observation files used have to be fairly clean.

3.47 *WhereSat*

3.47.1 Overview

This application uses input ephemeris to compute the predicted location of a satellite. The Earth-centered, Earth-fixed (ECEF) position of the satellite is reported. Optionally, the topocentric coordinates–azimuth, elevation, and range–can be generated. The user can specify the time interval between successive predictions. Also the output can be generated in a format easily imported into numerical packages.

3.47.2 Usage

WhereSat

Required Arguments		
Short Arg.	Long Arg.	Description
-e	–eph-files=ARG	Ephemeris source file(s). Can be RINEX nav, SP3, or FIC.
Optional Arguments		
Short Arg.	Long Arg.	Description
-h	–help	Print help usage.
-u	–position=ARG	Antenna position in ECEF (x,y,z) coordinates. Format as string: "X Y Z". used to give user-centered data (SV range, azimuth, and elevation) when SV is in view.
	–start=ARG	Ignore data before this time. Format as string: "MO/DD/YYYY HH:MM:SS".
	–end=ARG	Ignore data after this time. Format as string: "MO/DD/YYYY HH:MM:SS".
-f	–time-format=ARG	CommonTime format specifier used for times in the output. The default is "%4Y %3j %02H:%02M:%4.1f".
-p	–prn=NUM	Which SVs to analyze. Repeat option for multiple satellites. If this option is not specified, all ephemeris data will be processed.
-t	–time=NUM	Time increment in seconds for ephemeris calculation. Default is 900 seconds (15 minutes).

3.47.3 Examples

```
> wheresat -e nav/s121001a.08n
Scanning over prnSet.
Scan complete, size = 0
Scanning over PRNs indices.
Scan complete, size = 32
Set timeFormat to %02m/%02d/%04Y %02H:%02M:%04.1f
positionOpt has count = 0
# time, PRN, X(m), Y(m), Z(m), Clock Correction(ms)
File read by EphReader.
2454466 86384000 0.000000000000000 GPS
2454466 86384000 0.000000000000000 GPS
2454468 07200000 0.000000000000000 GPS
```

```
01/01/2008 00:14:44.0 2  21747436.312  -9257796.753 -11975305.200 -0.000019     -       -
-
01/01/2008 00:14:44.0 4  18510072.147   1615384.170 -19198127.464 0.000013      -       -
-
01/01/2008 00:14:44.0 10 22497303.175  -3072267.608  13925841.366 -0.000017     -       -
-
01/01/2008 00:14:44.0 11 -8870210.278  23588157.454  -8007606.591  0.000015     -       -
-
01/01/2008 00:14:44.0 12  5936939.408 -13991621.825 -21687667.199 0.000003      -       -
-
01/01/2008 00:14:44.0 13  6207240.073  25803219.367   1294271.894 -0.000008     -       -
-
01/01/2008 00:14:44.0 17 14616066.283  16715096.969 -14666158.339 -0.000006     -       -
-
...
```

Index