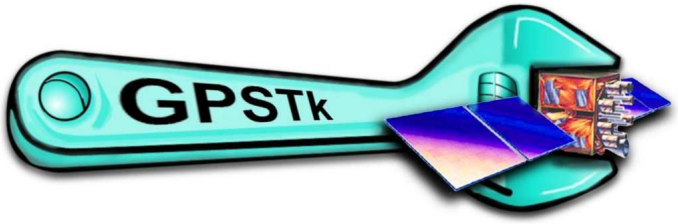

The GPS Toolkit

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

Geophysics Department, Helsinki University of Technology

January 4, 2007
GPSTk Revision 351

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 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 ibrary and a set of applications. The 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 29 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 layers 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 effects 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 Ephemeredes

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 ephemeredes, 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 GPS 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 ephemeredes 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 system 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] Barbara Wiley, David Craig, Dennis Manning, John Novak, Randall Taylor, and Leonard Weingarth. NGA's Role in GPS. In *Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Fort Worth, Texas, September 2006.

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 supporting 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 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, Charater (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.

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	read Rinex files and checks it 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	computes range residuals or zero baseline differences	`reszilla -o arl210.06o -e arl2100.06n`
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 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 2.1: GPSTk Applications, categorized, with execution examples.

2.5 *calgps*

2.5.1 Overview

This application generates a dual GPS and Julian calendar. 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.

2.5.2 Usage

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 year	–specific-year=NUM	Prints a GPS calendar for the entire specified year.

2.5.3 Examples

```
> calgps -3
```

```
> calgps -Y 1998


                    Jan 1998
938                             1-001  2-002  3-003
939   4-004  5-005  6-006  7-007  8-008  9-009 10-010
940  11-011 12-012 13-013 14-014 15-015 16-016 17-017
941  18-018 19-019 20-020 21-021 22-022 23-023 24-024
942  25-025 26-026 27-027 28-028 29-029 30-030 31-031

                    Feb 1998
943   1-032  2-033  3-034  4-035  5-036  6-037  7-038
944   8-039  9-040 10-041 11-042 12-043 13-044 14-045
945  15-046 16-047 17-048 18-049 19-050 20-051 21-052
946  22-053 23-054 24-055 25-056 26-057 27-058 28-059

                    Mar 1998
. . .
```

2.5.4 Notes

If multiple options are given only the first is considered.

2.6 *DiscFix*

2.6.1 Overview

The application reads a data file containing dual-frequency pseudorange and phase measurements and finds and fixes discontinuities in the phase; output is a list of editing commands (for use with PRGM RinexEdit); the program will also (optional) write out the raw (uncorrected) data to a RINEX file.

2.6.2 Usage

DiscFix

Required Arguments

Short Arg.	Long Arg.	Description
-i	–inputfile	Input (Rinex obs) file(s)

Optional Arguments

-f		file containing more options
-d	–directory	Directory of input file(s)
	–decimate	Decimate data to time interval dt
	–EpochBeg	Start time, arg is of the form YYYY,MM,DD,HH,Min,Sec
	–GPSBeg	Start time, arg is of the form GPSweek,GPSsow
	–EpochEnd	End time, arg is of the form YYYY,MM,DD,HH,Min,Sec
	–GPSEnd	End time, arg is of the form GPSweek,GPSsow
	–CA	Use C/A code pseudorange if P1 is not available
	–DT	Time interval (s) of data points (needed for -Ps only)
	–Gap	Time (s) of largest allowed gap within pass
	–Points	Minimum number of points needed to process a pass
	–XPRN	Exclude this satellite (prn may be only <system>)
	–SVonly	Process this satellite ONLY
	–Log	Output log file name (df.log)
	–Err	Output error file name (df.err)
	–Out	Output (editing commands) file name (df.out)
	–RinexFile	Output Rinex obs 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
	–Smooth	Smooth pseudorange and debias phase and output both in place of raw
	–SmoothPR	Smooth pseudorange and debias phase but replace only raw pseudorange
	–SmoothPH	Smooth pseudorange and debias phase but replace only raw phase
	–CAOut	Output C/A code in Rinex
	–DOut	Output Doppler in Rinex
	–verbose	print extended output (NB –DCDebug,7 =¿ all debugging output)
-h	–help	print syntax and quit.

2.6.3 Examples

```
> DiscFix --dt 1.5 --inputfile arl2800.06o

DiscFix, part of the GPS ToolKit, Ver 4.0 2/28/06, Run 2007/01/04 15:36:37
DiscFix is writing to log file df.log
DiscFix is writing to output file df.out
DiscFix timing: 0.960 seconds.
```

2.7 *ephdiff*

2.7.1 Overview

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

2.7.2 Usage

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

2.7.3 Examples

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

Data in fic06.187 not found in arl2800.06n:

********************************************************************************
Broadcast Ephemeris (Engineering Units)

PRN : 1

             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

 . . .
```

2.7.4 Notes

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

2.8 *fic2rin*

2.8.1 Overview

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

2.8.2 Usage

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

2.8.3 Examples

```
> fic2rin fic06.187 rin1870.06
```

```
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
 . . .
```

2.8.4 Notes

2.9 *ficacheck ficcheck*

2.9.1 Overview

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

2.9.2 Usage

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 = "beginning of time")
-e	–end-time=TIME	End of time range to compare (default = "end of time")

ficacheck usage: ficacheck [options] ¡FICA file¿
ficcheck usage: ficcheck [options] ¡FIC file

2.9.3 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:928
location 1:src/FFStream.cpp:127
location 2:src/FFStream.hpp:174
location 3:src/FFStream.hpp:174
```

2.9.4 Notes

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

2.10 *ficafic ficfica*

2.10.1 Overview

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

2.10.2 Usage

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

2.10.3 Examples

```
> ficfica fic06.187 fica06.187
```

```
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
 . . .
```

2.10.4 Notes

2.11 *ficdiff*

2.11.1 Overview

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

2.11.2 Usage

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 = "beginning of time")
-e	–end-time=TIME	End of time range to compare (default = "end of time")

ephdiff usage: ficdiff [options] fic1 fic2

2.11.3 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:
 . . .
```

2.11.4 Notes

2.12 *IonoBias*

2.12.1 Overview

The application will open and read several preprocessed RINEX obs 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.

2.12.2 Usage

IonoBias

Optional Arguments		
	–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		
	–navdir	Path of navigation file(s)
	–nav	Navigation (Rinex Nav OR SP3) file(s)
Output		
	–datafile	Data (AT) file name, for output and/or input
	–log	Output log file name
	–biasout	Output satellite+receiver biases file name
Time limits		
	–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		
	–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

	–XSat	Exclude this satellite (¡sat¿ may be ¡system¿ only)
-v	–verbose	print extended output info.
-d	–debug	print extended output info.
-h	–help	print syntax and quit.

2.12.3 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
01010110110000111111101110111011110
01010110110000111111101110111011110
01010010110000111111101110111011110
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
```

2.12.4 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.

2.13 *mdp2fic mdp2rinex*

2.13.1 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.

2.13.2 Usage

mdp2fic

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 to.
-t	–thinning=ARG	A thinning factor for the data, specified in seconds between points. Default: none.
-c	–12c=ARG	Enable output of L2C data in C2.

2.13.3 Examples

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

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

2.13.4 Notes

2.14 *mdptool*

2.14.1 Overview

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

2.14.2 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
-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. Sorry.
-l	–timeSpan=NUM	How much data to process, in seconds
	–startTime=TIME	Ignore data before this time. (%4Y/%03j/%02H:%02M:%05.2f)
	–stopTime=TIME	Ignore any data after this time

2.14.3 Examples

2.14.4 Notes

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

2.15 *mergeFic*

2.15.1 Overview

The applications merge multiple FIC files into a single FIC file.

2.15.2 Usage

mergeFIC

Required Arguments

Short Arg.	Long Arg.	Description
-i	–input=ARG	An input RINEX Obs file, can be repeated as many times as needed.
-o	–output=ARG	Name for the merged output RINEX Obs 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

2.15.3 Examples

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

2.15.4 Notes

2.16 *mergeRinObs mergeRinNav mergeRinMet*

2.16.1 Overview

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

2.16.2 Usage

mergeRinObs

Required Arguments

Short Arg.	Long Arg.	Description
-i	–input=ARG	An input RINEX Obs file, can be repeated as many times as needed.
-o	–output=ARG	Name for the merged output RINEX Obs 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 *mergeRinNav* have the same usage.

2.16.3 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
```

2.16.4 Notes

2.17 *navdmp*

2.17.1 Overview

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

2.17.2 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

2.17.3 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

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

2.17.4 Notes

2.18 *NavMerge*

2.18.1 Overview

The application merges RINEX Nav files into a single file.

2.18.2 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. Timetags have the form year,mon,day,HH,min,sec OR GPSweek,sow

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

2.18.3 Examples

```
> NavMerge -os081213-214.99n s081213a.99n s081214a.99n

Output file name is s081213-214.99n
Read 200 ephemerides from file s081213a.99n
Read 197 ephemerides from file s081214a.99n
Read 397 total ephemerides.
text 0:Unknown RINEX version: 2.11
text 1:Make sure to set the version correctly.
text 2:In record 0
text 3:In file s081213-214.99n
text 4:Near file line 0
location 0:src/RinexNavHeader.cpp:87
location 1:src/FFStream.cpp:228
location 2:src/FFStream.hpp:174
location 3:src/FFStream.hpp:174
```

2.18.4 Notes

NavMerge will also correct the output data when the GPS full week number is inconsistent with the epoch time.

2.19 *navsum*

2.19.1 Overview

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

2.19.2 Usage

navsum

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

2.19.3 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
 . . .
```

2.19.4 Notes

2.20 *novaRinex*

2.20.1 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.

2.20.2 Usage

NavMerge

Required Arguments

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

Optional Arguments

-f		Name of file containing more options ('#' to EOL : comment)
	–dir	Directory in which to find input file (defaults to ./)
	–obs	Rinex observation output file (RnovaRinex.obs)
	–nav	Rinex navigation output file (RnovaRinex.nav)

Output RINEX Header Fields

	–noHDopt	If present, do not fill optional records in the output Rinex header
	–HDp	Set output Rinex header 'program' field ('novaRinex v1.1 2/06')
	–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

	–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

	–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
-d	–debug	print extended output info

2.20.3 Examples

2.20.4 Notes

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

2.21 *poscvt*

2.21.1 Overview

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

2.21.2 Usage

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

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

2.21.3 Examples

```
> poscvt --ecef="-4346070.69263 4561978.26297 803.498856837"


   ECEF (x,y,z) in meters           -4346070.6926 4561978.2630 803.4989
   Geodetic (llh) in deg, deg, m    0.00735641 133.61157352 -77345.2412
   Geocentric (llr) in deg, deg, m  0.00730656 133.61157352 6300791.7584
   Spherical (tpr) in deg, deg, m   89.99269344 133.61157352 6300791.7584
```

```
> poscvt -l

 %X %Y %Z (cartesian or ECEF in kilometers)
%x %y %z (cartesian or ECEF in meters)
%a %l %r (geocentric lat,lon,radius, longitude E, radius in meters)
%A %L %h (geodetic lat,lon,height, longitude E, height in meters)
%a %w %R (geocentric lat,lon,radius, longitude W, radius in kilometers)
%A %W %H (geodetic lat,lon,height, longitude W, height in kilometers)
%t %p %r (spherical theta, phi, radius, degrees and meters)
%T %P %R (spherical theta, phi, radius, radians and kilometers)
```

```
> poscvt --ecef="-4346070.69263 4561978.26297 803.498856837" -F "\%A \%L \%h"

\0.007356 \133.611574 \-77345.241247
```

2.21.4 Notes

2.22 *PRSolve*

2.22.1 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 obs file with the position solutions in auxiliary header blocks.

2.22.2 Usage

navdmp

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		
-f		File containing more options
	–obsdir	Directory of input observation file(s)
	–navdir	Directory of input navigation file(s)
	–decimate	Decimate data to time interval dt
	–EpochBeg	Start time, arg is of the form YYYY,MM,DD,HH,Min,Sec
	–GPSBeg	Start time, arg is of the form GPSweek,GPSsow
	–EpochEnd	End time, arg is of the form YYYY,MM,DD,HH,Min,Sec
	–GPSEnd	End time, arg is of the form GPSweek,GPSsow
	–CA	Use C/A code pseudorange if P1 is not available
Optional Arguments: Configuration		
	–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)
	–MinElev	Minimum elevation angle (deg) (only if –PosXYZ)
	–XPRN	Exclude this satellite.
	–Trop <model,T,P,H>	Trop model (one of BL,SA,NB,GG,GGH (cf.GPSTk)), with OPTIONAL weather Temp(C),Press(mb),RH(%)
Optional Arguments: Output		
	–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 DayTime) (default: %4F %10.3g)

Optional Arguments: RINEX Output

	–RinexFile	Output Rinex obs 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

	–verbose	Print extended output.
	–debug	Print very extended output.
-h	–help	Print syntax and quit.

2.22.3 Examples

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

PRSolve, part of the GPSTK ToolKit, Ver 1.9 9/06, Run 2007/01/04 15:36:50
Opened log file prs.log
Weighted average RAIM solution for all files
   918129.320229 -4346071.108765 4561977.869659
Covariance of RAIM solution for all files
        0.000150       -0.000061        0.000058
       -0.000061        0.000427       -0.000248
        0.000058       -0.000248        0.000493
PRSolve timing: 2.450 seconds.
```

```
> PRSolve -o arl2800.06o -n arl2800.06n --EpochBeg 2006,1,1,00,00,00 --EpochEnd 2006,1,1,12,00,00

PRSolve, part of the GPSTK ToolKit, Ver 1.9 9/06, Run 2007/01/04 15:36:55
Opened log file prs.log
Weighted average RAIM solution for all files
   918129.968984 -4346071.600388 4561978.175321
Covariance of RAIM solution for all files
        0.000315       -0.000130        0.000155
       -0.000130        0.000918       -0.000516
        0.000155       -0.000516        0.001041
PRSolve timing: 1.130 seconds.
```

2.22.4 Notes

2.23 *ResCor*

2.23.1 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.

2.23.2 Usage

ResCor

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

	–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

Residual/Correction computation

	–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)
	–SVonly <prn>	Process this satellite ONLY

Output files

	–Log <file>	Output log file name (rc.log)
	–Err <file>	Output error file name (rc.err)

Help

	–verbose	Print extended output
	–debug	Print debugging information.
-h	–help	Print syntax and quit.

```
Rinex Editor commands:
 ================================================================================
 Commands begin with a '-' or '/', followed by an identifier, then data fields.
 Fields beyond the initial 2- or 3-character identifier are comma delimited.
 <SV> gives a satellite; SV=<PRN><System(optional)> eg. 19G or 19 = PRN 19 GPS.
 <System> is a single character (G=GPS, R=GLONASS, T=Transit, S=Geosynchronous).
 <OT> gives a Rinex observation type, e.g. L1 or P2 (case sensitive).
 <time> gives a time; time=<week,sow> OR time=<year,mon,day,hour,min,second>.
```

File I/O

-IF<file>	Input Rinex observation file name (required)
-ID<dir>	Directory in which to find input file
-OF<file>	Output Rinex file name (required, or -OF<file>,<time>)
-OF<f>,<time>	At time=<time>, close output file and open another named <f>
-OD<dir>	Directory in which to put output file(s)

Output Rinex header fields

-HDf	If present, fill optional records in the output Rinex header (NB EditObs() and EditFile() will do this, but NOT EditHeader().)
-HDp<program>	Set output Rinex header 'program' field
-HDo<observer>	Set output Rinex header 'observer' field
-HDa<agency>	Set output Rinex header 'agency' field
-HDm<marker>	Set output Rinex header 'marker' field
-HDn<number>	Set output Rinex header 'number' field
-HDc<comment>	Add comment to output Rinex header (more than one allowed).
-HDdc	Delete all comments in output Rinex header (NB -HDdc cannot delete comments created by *subsequent* -HDc commands)
-AO<OT>	Add observation type OT to header and observation data

General edit commands

-TB<time>	Begin time: reject data before this time (also used for decimation)
-TE<time>	End time: reject data after this time
-TT<dt>	Tolerance in comparing times, in seconds (default=1ms)
-TN<dt>	Decimate data to epochs = Begin + integer*dt (within tolerance)

```
 Specific edit commands:
 =======================
 (Generally each '+' command (e.g DA+,<time>) has a corresponding '-' command,
  and vice-versa; if not, End-of-file or Begin-of-file is assumed.
  Note commands at one time are applied AFTER other commands of the same type.)
```

Delete Command	
-DA+<time>	Delete all data beginning at this time
-DA-<time>	Stop deleting data at this time
-DO<OT>	Delete observation type OT entirely (including in header)
-DS<SV>	Delete all data for satellite SV entirely (SV may be system only)
-DS<SV>,<time>	Delete all data for satellite SV at this single time (only)
-DA-<time>	Stop deleting data at this time
-DO<OT>	Delete observation type OT entirely (including
-DS<SV>	Delete all data for satellite SV entirely (SV may be system only)
-DS<SV>,<time>	Delete all data for satellite SV at this single time (only)
-DS+<SV>,<time>	Delete all data for satellite SV beginning at this time
-DS-<SV>,<time>	Stop deleting all data for satellite SV at this time (NB DS commands with SV=system (only) delete all satellites of that system.)
-DD<SV,OT,t>	Delete a single Rinex data(SV,OT,t) at time <t>
-DD+<SV,OT,t>	Delete all (SV,OT) data, beginning at time <t>
-DD-<SV,OT,t>	Stop deleting all (SV,OT) data at time <t> (NB deleting data for one OT means setting it to zero - here and in Rinex)
Set Commands	
-SD<SV,OT,t,d>	Set data(SV,OT,t) to <d> at time <t>
-SS<SV,OT,t,s>	Set ssi(SV,OT,t) to <s> at time <t>
-SL+<SV,OT,t,l>	Set all lli(SV,OT,t) to <l> at time <t>
-SL-<SV,OT,t,l>	Stop setting lli(SV,OT,t) to <l> at time <t> (',<l>' is optional)
-SL<SV,OT,t,l>	Set lli(SV,OT,t) to <l> at the single time <t> (only) (NB SL commands with SV=system (only) modify all satellites of that system.)
Bias Commands	
	(NB. BD commands apply only when data is non-zero, unless -BZ appears)
-BZ	Apply bias data commands (BD) even when data is zero
-BD<SV,OT,t,d>	Add the value of <d> to data(SV,OT,t) at time <t>
-BD+<SV,OT,t,d>	Add value of <d> to data(SV,OT) beginning at time <t>
-BD-<SV,OT,t,d>	Stop adding <d> to data(SV,OT) at time <t> (',<d>' optional)
-BS<SV,OT,t,s>	Add the value of <s> to ssi(SV,OT,t) at time <t>
-BL<SV,OT,t,l>	Add the value of <l> to lli(SV,OT,t) at time <t>

```
The list of available extended Rinex obs 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
```

2.23.3 Examples

2.23.4 Notes

2.24 *reszilla*

2.24.1 Overview

Reszilla is an application that computes 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.

2.24.2 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.

The ORD Options:

```
-n, --search-near
--svtime
--check-obs
--omode
```

2.24.3 Double Difference Residuals

While many double differences exist, reszilla computes an 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 computing their observation. To remove this error, an estimate of the clock offset between the two receivers is need. Reszilla can get this estimate in one of two ways; estimates this by computing a clock estimate for each receiver as described under the

ORD section or reading 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 slight of hand. Do not delve into this or even look too closely at the details or you will be sullied.

DD Options:

```
--ddmode
--min-arc-time
--min-arc-gap
--min-arg-length
```

2.24.4 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).

Options:

```
-o, --obs1
-2, --obs2
-e, --eph
-w, --weather
-c, --clock-from-rinex
-m, --msid
--msc
```

2.24.5 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 a 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.

Output Options:

```
--keep-unhealthy
-s, --no-stats
--cycle-slips
-r, --raw-output
-t, --time-format
--clock-est
-b, --elev-bin
--sigma
-v, --verbosity
```

2.24.6 Usage

Required Arguments

Short Arg.	Long Arg.	Description
-o	–obs1=ARG	Observation data file name. If this option is specified more than once the contents of all files will be used.

Optional Arguments

Short Arg.	Long Arg.	Description
-h	–help	Generates help and usage.
-2	–position=ARG	Second receiver's observation data file name. Only used when computing a double difference. If this option is specified more than once the contents of all the files will be used.
	–msc=ARG	Station coordinate file.
-e	–ephemeris=ARG	Ephemeris data file name (either broadcast in RINEX nav, broadcast in FIC, or precise in SP3).
-w	–weather	Weather data file name (RINEX met format only).
-n	–search-near	Use BCEphemeris.searchNear()
-c	–clock-from-rinex	Use the receiver clock offset from the rinex obs data.
	–svtime	Observation data is in SV time frame. The default is RX time frame.
	–check-obs	Report data rate, order of data, data present, data gaps.
	–keep-unhealthy	Keeps unhealthy SVs in the statistics, default is to toss.
-s	–no-stats	Don't compute output the statistics.
	–cycle-slips	Output a list of cycle slips.
-r	–raw-output=ARG	Dump the computed residuals/ords into specified file. If '-' is given as the file name, the output is sent to standard output. The default is to not otput the raw residuals.

	–start-time=TIME	Ignore obs data prior to this time in the analysis. The time is specified using the format %4Y/%03j/%02H:%02M:%05.2f. The default value is to start with the first data found.
	–stop-time=TIME	Ignore obs data after to this time in the analysis. The time is specified using the format %4Y/%03j/%02H:%02M:%05.2f. The default value is to process all data.
-t	–time-format=ARG	Daytime format specifier used for the timestamps in the raw output. The default is "%Y %3j %02H:%02M:%04.1f". If this option is specified with the format as "s", the format "%Y %3j %7.1s" is used. If this option is specified with the format as "s", the format "%Y %3j %02H:%02M:%02S" is used.
	–omode=ARG	ORD mode: P1P2, C1P2, C1, P1, P2. The default is p1p2
	–clock-est	Compute a linear clock estimate.
	–ddmode=ARG	Double difference residual mode: none, sv, or c1p2. The default is sv.
	–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=NUM	The minimum number of epochs that can be considered an arc. The default value is 5 epochs.
-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".
	–sigma=NUM	Multiplier for sigma stripping used in computation of statistics on the raw residuals. The default value is 6.
-v	–verbosity=NUM	How much detail to provide about intermediate steps.
	0	nothing but the results
	1	Output status before potentially time consuming operations (default)
	2	more details about each step and the options chosen
	3	add the reasons for editing data
	4	dump intermediate values for each epoch (can be QUITE verbose)

```
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
```

2.24.7 Examples

```
reszilla --omode=p1 --svtime --msc=mscoords.cfg -m 85401 -o asm2004.138 -e s011138a.04n
```

2.24.8 Notes

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

2.25 *rmwcheck rnwcheck rowcheck*

2.25.1 Overview

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

2.25.2 Usage

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 = "beginning of time")
-e	–end-time=TIME	End of time range to compare (default = "end of time")

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

2.25.3 Examples

```
> rnwcheck -t "08/01/2006 12:00:00" -e "08/01/2006 15:00:00" s081213a.99n

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

2.25.4 Notes

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

2.26 *rmwdiff rnwdiff rowdiff*

2.26.1 Overview

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

2.26.2 Usage

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 = "beginning of time")
-e	–end-time=TIME	End of time range to compare (default = "end of time")

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>

2.26.3 Examples

2.26.4 Notes

2.27 *RinexDump*

2.27.1 Overview

The application reads a RINEX file and dumps the obervation data for the given satellite(s) to the standard output.

2.27.2 Usage

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

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

2.27.3 Examples

```
> RinexDump 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
1378 259290.000 G03 -4175846.973 0 3 -3253901.944 0 3 23733104.211 0 0
1378 259320.000 G03 -4286607.460 0 4 -3340208.647 0 3 23712026.249 0 0
1378 259350.000 G03 -4397272.869 0 4 -3426441.227 0 3 23690967.159 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
23733104.928 0 0 23733102.480 0 0     19.900 0 0      21.600 0 0
23712027.682 0 0 23712024.790 0 0     24.200 0 0      19.300 0 0
23690968.861 0 0 23690965.837 0 0     25.600 0 0      19.900 0 0

 . . .
```

2.27.4 Notes

MATLAB and Octave can read the purely numeric output.

2.28 *rinexpvt*

2.28.1 Overview

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

2.28.2 Usage

navdmp

Required Arguments

Short Arg.	Long Arg.	Description
-o	–obs-file=ARG	RINEX obs 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 Nav 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 Met 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)
-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.

2.28.3 Examples

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

2006 1 1 00 00 00 918128.1413 -4346066.38713 4561976.84865 322.333995519
2006 1 1 00 00 30 918128.209212 -4346067.60732 4561976.93485 323.041856353
2006 1 1 00 01 00 918128.302764 -4346068.04452 4561977.21068 323.429649855
2006 1 1 00 01 30 918128.391428 -4346068.3532 4561977.38928 323.717577661
2006 1 1 00 02 00 918128.50273 -4346068.53469 4561977.48638 323.86573351
2006 1 1 00 02 30 918128.529272 -4346068.41506 4561977.46288 323.78986994
2006 1 1 00 03 00 918128.646582 -4346068.55693 4561977.52889 323.955585289
2006 1 1 00 03 30 918128.740209 -4346068.77352 4561977.6377 324.13232439
2006 1 1 00 04 00 918128.739294 -4346068.83903 4561977.68601 324.180075896
2006 1 1 00 04 30 918128.781829 -4346068.85625 4561977.77165 324.239920157
2006 1 1 00 05 00 918128.861036 -4346069.05268 4561977.91535 324.454651606
2006 1 1 00 05 30 918128.933265 -4346069.40007 4561978.12808 324.786489416
2006 1 1 00 06 00 918128.950514 -4346069.25246 4561978.14827 324.733986098
```

```
2006 1 1 00 06 30 918128.960248 -4346069.24879 4561978.11298 324.748810797
2006 1 1 00 07 00 918128.976853 -4346069.3422 4561978.17787 324.858597826
 . . .
```

```
> rinexpvt -o arl2800.06o -n arl2800.06n -m arl2800.06m

2006 1 1 00 00 00 918128.1413 -4346066.38713 4561976.84865 322.333995519
2006 1 1 00 00 30 918128.209212 -4346067.60732 4561976.93485 323.041856353
2006 1 1 00 01 00 918128.401075 -4346068.40185 4561977.50754 323.99086869
2006 1 1 00 01 30 918128.488498 -4346068.70699 4561977.68361 324.275285634
2006 1 1 00 02 00 918128.598571 -4346068.88502 4561977.77824 324.42000745
2006 1 1 00 02 30 918128.623895 -4346068.76203 4561977.75232 324.340785521
2006 1 1 00 03 00 918128.739997 -4346068.90062 4561977.81596 324.503217171
2006 1 1 00 03 30 918128.832428 -4346069.114 4561977.92245 324.676746145
2006 1 1 00 04 00 918128.830326 -4346069.1764 4561977.9685 324.721360094
2006 1 1 00 04 30 918128.871684 -4346069.19058 4561978.05191 324.778138464
2006 1 1 00 05 00 918128.949723 -4346069.38404 4561978.19345 324.989874831
2006 1 1 00 05 30 918129.020728 -4346069.7283 4561978.40383 325.318381098
2006 1 1 00 06 00 918129.036829 -4346069.57789 4561978.42195 325.263023987
2006 1 1 00 06 30 918129.045424 -4346069.57149 4561978.38464 325.275063272
2006 1 1 00 07 00 918129.0609 -4346069.66224 4561978.44755 325.382132551
 . . .
```

2.28.4 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 3 files must be included, the previous day, the current day and the next day.

2.29 *rinexthin*

2.29.1 Overview

This application decimates an input RINEX observation file to a specified data rate.

2.29.2 Usage

navdmp

Required Arguments

Short Arg.	Long Arg.	Description
-f	–filename=ARG	RINEX obs file to be thinned.
-s	–Seconds=NUM	The desired data rate.
-o	–filename=ARG	RINEX obs file with thinned obs.

2.29.3 Examples

```
> rinexthin -f arl2800.06o -s 60 -o arl2800thin.06o

Obs read:   2880
Obs written: 1440
```

2.29.4 Notes

2.30 *RinSum*

2.30.1 Overview

The application reads a RINEX file and summarizes it content.

2.30.2 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.
	–EpochBeg	Start time, arg is of the form YYYY,MM,DD,HH,Min,Sec
	–GPSBeg	Start time, arg is of the form GPSweek,GPSsow
	–EpochEnd	End time, arg is of the form YYYY,MM,DD,HH,Min,Sec
	–GPSEnd	End time, arg is of the form GPSweek,GPSsow
-h	–help	print syntax and quit.
-d	–debug	print debugging info.

2.30.3 Examples

```
> RinSum -i data_set/s081213a.99o --EpochBeg 2006,08,1,12,0,0'

++++++++++++++ RinSum summary of Rinex obs file data_set/s081213a.99o ++++++++++++++
Rinex header:
-------------------------------- REQUIRED --------------------------------
Rinex Version 2.10, File type Observation, System G (GPS).
Prgm: RinexObsWriter, Run: 11-14-01 10:04:27, By: NIMA
Marker name: 85408.
Obs'r : Monitor Station, Agency: NIMA
Rec#: 1, Type: ZY12, Vers:
Antenna # : 85408, Type : AshTech Geodetic 3
Position (XYZ,m) : (-740289.7851, -5457071.6555, 3207245.8294).
Antenna offset (ENU,m) : (0.0000, 0.0000, 0.0000).
Wavelength factors (default) L1:1, L2: 1.
Observation types (7) :
 Type #0 = L1 L1 Carrier Phase (L1 cycles).
 Type #1 = L2 L2 Carrier Phase (L2 cycles).
 Type #2 = C1 C/A-code pseudorange (meters).
 Type #3 = P1 Pcode L1 pseudorange (meters).
 Type #4 = P2 Pcode L2 pseudorange (meters).
 Type #5 = D1 Doppler Frequency L1 (Hz).
 Type #6 = D2 Doppler Frequency L2 (Hz).
Time of first obs 1999/08/01 00:00:00.0000000 GPS
```

```
(This header is VALID 2.1 Rinex.)
-------------------------------- OPTIONAL ----------------------------------
Comments (3) :
The AS bit flag is set if receiver is in Z mode
Signal to Noise ratio information is omitted
This file contains SMOOTHED obs data
------------------------------- END OF HEADER ------------------------------

 WARNING: Computed first time does not agree with header

 Computed interval is 0.00
 Computed first epoch is -4713/01/01 00:00:00.0000000
 Computed last epoch is 1999/08/01 23:59:30.0000000
 There were 0 epochs (-0.00% of -2147483647 possible epochs in this timespan) and 0 inline header block

         Summary of data available in this file: (Totals are based on times and interval)
PRN/OT:    L1    L2    C1    P1    P2    D1    D2 Total Begin - End time
TOTAL       0     0     0     0     0     0     0

 WARNING: ObsType L1 should be deleted from header.

 WARNING: ObsType L2 should be deleted from header.

 WARNING: ObsType C1 should be deleted from header.

 WARNING: ObsType P1 should be deleted from header.

 WARNING: ObsType P2 should be deleted from header.

 WARNING: ObsType D1 should be deleted from header.

 WARNING: ObsType D2 should be deleted from header.

++++++++++++++ End of RinSum summary of data_set/s081213a.99o ++++++++++++++
```

2.30.4 Notes

2.31 *rtAshtech*

2.31.1 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.

2.31.2 Usage

rtAshtech
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
-p	–port=ARG	Serial port to use
-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

2.31.3 Examples

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

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

2.31.4 Notes

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

2.32 *TECMaps*

2.32.1 Overview

The application will open and read several preprocessed RINEX obs files (containing obs types EL,AZ,VR—SR) and use the data to create maps of the Total Electron Content (TEC).

2.32.2 Usage

TECMaps

Required Arguments

–input	Input Rinex obs file name(s)

Optional Arguments

-f	file containing more options

Reference station position (one required)

–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

–navdir	Path of navigation file(s)
–nav	Navigation (Rinex Nav OR SP3) file(s)

Output

–log	Output log file name

Time limits

–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

–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 information
–Title2	Second title information
–BaseName	Base name for output files (a)
–DecorrError	Decorrelation error rate in TECU/1000km (3)
–Biases	File containing estimated sat+rx biases (Prgm IonoBias)
–ElevThresh	Minimum elevation (6 deg)
–MinAcqTime	Minimum acquisition time (0 sec)
–FlatFit	Flat fit type (default)
–LinearFit	Linear fit type
–IonoHeight	Ionosphere height (km)

Grid

–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	Number of latitude grid points (40)
–NumLon	Number of longitude grid points (40)
–BeginLat	Beginning latitude (21 deg)
–BeginLon	Beginning longitude (230 deg E)
–DeltaLat	Grid spacing in latitude (0.25 deg)
–DeltaLon	Grid spacing in longitude (1.0 deg)

Other options

–XSat	Exclude this satellite (¡sat¿ may be ¡system¿ only)

Help

-v	–verbose	print extended output info.
-d	–debug	print extended output info.
-h	–help	print syntax and summary of input, then quit.

2.32.3 Examples

2.32.4 Notes

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

2.33 *timeconvert*

2.33.1 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).

2.33.2 Usage

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level
-v	–verbose	Increase verbosity
-h	–help	Print help usage
-c	–calendar=TIME	“Month(numeric) DayOfMonth Year”
-r	–rinex=TIME	”Month(numeric) DayOfMonth Year Hour:Minute:Second”
-R	–rinex-file=TIME	”Year(2-digit) Month(numeric) DayOfMonth Hour Minute Second”
-y	–doy=TIME	”Year DayOfYear SecondsOfDay”
-m	–mjd=TIME	”ModifiedJulianDate”
-o	–shortweekandsow=TIME	”10bitGPSweek SecondsOfWeek Year”
-z	–shortweekandzcounts=TIME	”10bitGPSweek ZCounts Year”
-f	–fullweekandsow=TIME	”FullGPSweek SecondsOfWeek”
-w	–fullweekandzcounts=TIMEo	”FullGPSweek ZCounts”
-u	–unixtime=TIME	”UnixSeconds UnixMicroseconds”
-Z	–fullZcounts=TIME	”fullZcounts”
-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

2.33.3 Examples

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


      Month/Day/Year               5/6/1985
      Hour:Min:Sec                 13:50:02
      Modified Julian Date         46191.576412037
      GPSweek DayOfWeek SecOfWeek  278 1 136202.000000
      FullGPSweek Zcount           278 90801
      Year DayOfYear SecondOfDay   1985 126 49802.000000
      Unix_sec Unix_usec           484235402 0
      FullZcount                   145842865
```

```
> timeconvert -o "1379 500 2006"


      Month/Day/Year               6/11/2006
      Hour:Min:Sec                 00:08:20
      Modified Julian Date         53897.005787037
      GPSweek DayOfWeek SecOfWeek  355 0 500.000000
```

```
       FullGPSweek Zcount           1379 333
       Year DayOfYear SecondOfDay   2006 162 500.000000
       Unix_sec Unix_usec           1149984500 0
       FullZcount                   186122573
```

```
> timeconvert -o "1379 500 2006 -a 86400"


       Month/Day/Year               6/11/2006
       Hour:Min:Sec                 00:08:20
       Modified Julian Date         53897.005787037
       GPSweek DayOfWeek SecOfWeek  355 0 500.000000
       FullGPSweek Zcount           1379 333
       Year DayOfYear SecondOfDay   2006 162 500.000000
       Unix_sec Unix_usec           1149984500 0
       FullZcount                   186122573
```

```
> timeconvert -w "1381 500" -s 200


       Month/Day/Year               6/25/2006
       Hour:Min:Sec                 00:09:10
       Modified Julian Date         53911.0063657407
       GPSweek DayOfWeek SecOfWeek  357 0 550.000000
       FullGPSweek Zcount           1381 366
       Year DayOfYear SecondOfDay   2006 176 550.000000
       Unix_sec Unix_usec           1151194150 0
       FullZcount                   187171182
```

2.33.4 Notes

If no arguments are given it will convert the current time to all formats.

2.34 *WhereSat*

2.34.1 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 generated in a format easily imported into numerical packages.

2.34.2 Usage

Required Arguments

Short Arg.	Long Arg.	Description
-b	–broadcast=ARG	Specify a RINEX navigation file. The user may enter multiple files.
-p	–prn=NUM	Specify which SV to analuze.

Optional Arguments

Short Arg.	Long Arg.	Description
-h	–help	Generates help and usage.
-u	–position=ARG	Specify antenna position in ECEF (x,y,z) coordinates as "X Y Z". Used to give user-centered data (SV range, azimuth & elevation).
-s	–start=ARG	Specify time to begin analysis as "MO/DD/YYYY HH:MM:SS". The default is the end of the file.
-e	–end=ARG	Specify time to end analysis as "MO/DD/YYYY HH:MM:SS". The default is the beggning of the file.
-o	–output-filename=ARG	Outputs results to a MATLAB readable file.
-t	–time=NUM	Specify time increment for ephemeris calculation in seconds. Default is 900 (15 min.)

2.34.3 Examples

```
> WhereSat -b aira1720.06n -p 2 -u "918129.01 -4346070.45 803.18"
  -s "06/21/2006 17:00:00" -e "06/21/2006 20:00:00" -t 1800

Antenna Position: 918129 -4.34607e+06 803.18
Navigation File: aira1720.06n
Start Time:      06/21/2006 17:00:00
End Time:        06/21/2006 20:00:00
PRN:             2

Prn 2 Earth-fixed position and clock information:

Date      Time(UTC)  X (meters)        Y (meters)        Z (meters)
================================================================================
06/21/2006 18:00:00 12758891.971859   18901201.616227    -14049016.596144
06/21/2006 18:30:00 12847888.097031   21541501.416411    -9315422.851798
```

```
06/21/2006 19:00:00 12843576.989405   23087218.618683    -3957280.515764
06/21/2006 19:30:00 12450313.769289   23516935.034029    1667186.089065

 . . .

   Clock Correc (s)
==================
    0.000007
    0.000007
    0.000007
    0.000007



Data for user reference frame:

Date      Time(UTC)  Azimuth       Elevation     Range to SV (m)
========================================================================
06/21/2006 18:00:00 130.596202    -43.242769     29627531.177821
06/21/2006 18:30:00 118.680085    -49.681012     29983796.522429
06/21/2006 19:00:00 102.845663    -53.888528     30169796.433699
06/21/2006 19:30:00 84.400419     -55.459042     30197072.648367

Calculated 4 increments for prn 2 .
```

2.34.4 Notes

2.35 *vecsol*

2.35.1 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. Ephemeris used are either broadcast or precise (SP3). Alternatively, also P code processing is 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.

2.35.2 Usage

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

RINEX Observation Files

The two arguments are names of RINEX observation files. The 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 co-ordinate differences between the baseline end points. If 0, solve for the absolute co-ordinates 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.

2.35.3 Examples

2.35.4 Notes

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

Index