WAYP★INT
PRODUCTS GROUP

A NovAtel Precise Positioning Product

RTKNav

USER GUIDE

RTKNav User Guide

Publication Number: OM-20000107

Revision Level: 2

Revision Date: 2008/08/01

This manual reflects RTKNav Software Version 4.00.

Proprietary Notice

Information in this document is subject to change without notice and does not represent a commitment on the part of NovAtel Inc. The software described in this document is furnished under a licence agreement or non-disclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or non-disclosure agreement.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of a duly authorized representative of NovAtel Inc.

The information contained within this manual is believed to be true and correct at the time of publication.

NovAtel, OEM4, Waypoint, GrafNav, GrafNet, Inertial Explorer, GPSolution, ProPak, OEMV RT-20 and RT-2 are registered trademarks of NovAtel Inc.

SPAN Technology, DL-4*plus*, ProPak-G2*plus*, ProPak-LB*plus* and PAC are trademarks of NovAtel Inc.

All other product or brand names are trademarks of their respective holders.

Table of Contents

List of Tables

Foreword

Congratulations!

Congratulations on purchasing a software package from NovAtel Inc.'s Waypoint Products Group.

RTKNav is a Windows-based suite of programs that provides real-time GPS data processing. This guide will help you install and navigate your software.

Scope

This guide contains information on the installation and operation of the Waypoint Products Group's RTKNav software package to allow you to effectively navigate and process GPS data. It is beyond the scope of this guide to provide details on service or repair, please see *Conventions and Customer Service* section on this page for customer support.

Prerequisites

To run Waypoint software packages, your personal computer must meet or exceed this minimum configuration:

Operating System
Windows 2000, XP, or Vista.

Hard Drive Space
55 MB of available space on the hard disk.

Memory
At least 256 MB of RAM is required.

Although previous experience with Windows is not necessary to use Waypoint software packages, familiarity with certain actions that are customary in Windows will assist in the usage of the program. This guide has been written with the expectation that you already have a basic familiarity with Windows.

Conventions and Customer Service

This guide covers the full performance capabilities of RTKNav real-time GPS data processing software. The conventions include the following:

> ✉ This is a notebox that contains important information before you use a command or log, or to give additional information afterwards.

The term "master" refers to the reference station and the base station.

The term "remote" refers to the rover or to a "station" in GrafNav.

This manual contains shaded boxes on the outside of the pages. These boxes contain procedures, screen shots and quick references.

If the software was purchased through a vendor, please contact them for support. Otherwise, for software updates and customer service, contact Waypoint using the following methods:

Call: (403) 295-4900
Fax: (403) 295-4901
E-mail: wpsupport@novatel.com
Web: http://www.novatel.com/
Write:NovAtel Inc.
    Customer Service Department
    1120-68 Avenue NE
    Calgary AB
    Canada, T2E 8S5

RTKNav 4.00 What's new!

Version 4.00 includes a new processing engine that uses NovAtel's Advance RTK. Fixed integer solutions are obtained quicker and more reliably and on longer baselines. The engine also supports GLONASS measurements.

Each baseline in RtGNSSx.DLL can now be configured separately, allowing different processing options for each baseline.

The RtGNSSx.DLL can now handle multiple moving baselines.

Accuracy is improved on base station data extrapolation. RTKNav data extrapolation also supports GLONASS measurements.

RTKNav supports real time Google Earth display.

SIOGPS.DLL now supports the RTCM 3 data format. This is a highly compressed format, ideal for minimal bandwidth use. SIOGPS.DLL uses the new GPB data format. The new GPB data format contains information for the signal to noise ratio, improved lock time tracking and data handling. SIOGPS.DLL has improved data rebroadcast. Data can now be rebroadcast without a delay and only critical records can be rebroadcast to minimize bandwidth consumption.

RTKNav/WLOG will configure NovAtel receivers for RTCM3. RTKNav/WLOG now supports the NovAtel USB interface. Support for the RtStatic module has been removed.

Software License

BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE PRODUCT, YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH THESE TERMS OF USE, DO NOT INSTALL, COPY OR USE THIS ELECTRONIC PRODUCT (SOFTWARE, FIRMWARE, SCRIPT FILES, OR OTHER ELECTRONIC PRODUCT WHETHER EMBEDDED IN THE HARDWARE, ON A CD OR AVAILABLE ON THE COMPANY WEB SITE) (hereinafter referred to as "Software").

1. **License:** NovAtel Inc. ("NovAtel") grants you a non-exclusive, non-transferable license (not a sale) to use the software subject to the limitations below. You agree not to use the Software for any purpose other than the due exercise of the rights and licences hereby agreed to be granted to you.

2. **Copyright:** NovAtel owns, or has the right to sublicense, all copyright, trade secret, patent and other proprietary rights in the Software and the Software is protected by national copyright laws, international treaty provisions and all other applicable national laws. You must treat the Software like any other copyrighted material and the Software may only be used on one computer at a time You may not copy the product manual or written materials accompanying the Software. No right is conveyed by this Agreement for the use, directly, indirectly, by implication or otherwise by Licensee of the name of NovAtel, or of any trade names or nomenclature used by NovAtel, or any other words or combinations of words proprietary to NovAtel, in connection with this Agreement, without the prior written consent of NovAtel.

3. **Patent Infringement:** NovAtel shall not be liable to indemnify the Licensee against any loss sustained by it as the result of any claim made or action brought by any third party for infringement of any letters patent, registered design or like instrument of privilege by reason of the use or application of the Software by the Licensee or any other information supplied or to be supplied to the Licensee pursuant to the terms of this Agreement. NovAtel shall not be bound to take legal proceedings against any third party in respect of any infringement of letters patent, registered design or like instrument of privilege which may now or at any future time be owned by it. However, should NovAtel elect to take such legal proceedings, at NovAtel's request, Licensee shall co-operate reasonably with NovAtel in all legal actions concerning this license of the Software under this Agreement taken against any third party by NovAtel to protect its rights in the Software. NovAtel shall bear all reasonable costs and expenses incurred by Licensee in the course of co-operating with NovAtel in such legal action.

4. **Restrictions:**

You may not:

(a) use the software on more than one computer simultaneously with exception of the Windows and WinCE data logging software which may be copied and used for each GPS receiver collected data simultaneously;

(b) distribute, transfer, rent, lease, lend, sell or sublicense all or any portion of the Software without the written permission of NovAtel;

(c) alter, break or modify the hardware protection key (dongle) thus disabling the software copy protection;

(d) modify or prepare derivative works of the Software;

(e) use the Software in connection with computer-based services business or publicly display visual output of the Software;

(f) implement DLLs and libraries in a manner that permits automated internet based post-processing (contact NovAtel for special pricing);

(g)transmit the Software over a network, by telephone or electronically using any means (except when downloading a purchased upgrade from the NovAtel web site); or

(h) reverse engineer, decompile or disassemble the Software.

You agree to keep confidential and use your best efforts to prevent and protect the contents of the Software from unauthorized disclosure or use.

5.   **Term and Termination:** This Agreement and the rights and licences hereby granted shall continue in force in perpetuity unless terminated by NovAtel or Licensee in accordance herewith. In the event that the Licensee shall at any time during the term of this Agreement: i) be in breach of its obligations hereunder where such breach is irremediable or if capable of remedy is not remedied within 30 days of notice from NovAtel requiring its remedy; then and in any event NovAtel may forthwith by notice in writing terminate this Agreement together with the rights and licences hereby granted by NovAtel. Licensee may terminate this Agreement by providing written notice to NovAtel. Upon termination, for any reasons, the Licensee shall promptly, on NovAtel's request, return to NovAtel or at the election of NovAtel destroy all copies of any documents and extracts comprising or containing the Software. The Licensee shall also erase any copies of the Software residing on Licensee's computer equipment. Termination shall be without prejudice to the accrued rights of either party, including payments due to NovAtel. This provision shall survive termination of this Agreement howsoever arising.

6.   **Warranty:** NovAtel does not warrant the contents of the Software or that it will be error free. The Software is furnished "AS IS" and without warranty as to the performance or results you may obtain by using the Software. The entire risk as to the results and performance of the Software is assumed by you. See product enclosure, if any for any additional warranty.

7.   **Indemnification:** NovAtel shall be under no obligation or liability of any kind (in contract, tort or otherwise and whether directly or indirectly or by way of indemnity contribution or otherwise howsoever) to the Licensee and the Licensee will indemnify and hold NovAtel harmless against all or any loss, damage, actions, costs, claims, demands and other liabilities or any kind whatsoever (direct, consequential, special or otherwise) arising directly or indirectly out of or by reason of the use by the Licensee of the Software whether the same shall arise in consequence of any such infringement, deficiency, inaccuracy, error or other defect therein and whether or not involving negligence on the part of any person.

8.   **Disclaimer and Limitation of Liability:**

(a) **THE WARRANTIES IN THIS AGREEMENT REPLACE ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NovAtel DISCLAIMS AND EXCLUDES ALL OTHER WARRANTIES. IN NO EVENT WILL NovAtel's LIABILITY OF ANY KIND INCLUDE ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, EVEN IF NOVATEL HAS KNOWLEDGE OF THE POTENTIAL LOSS OR DAMAGE.**

(b) NovAtel will not be liable for any loss or damage caused by delay in furnishing the Software or any other performance under this Agreement.

(c) NovAtel's entire liability and your exclusive remedies for our liability of any kind (including liability for negligence) for the Software covered by this Agreement and all other performance or non-performance by NovAtel under or related to this Agreement are to the remedies specified by this Agreement.

9.   **Governing Law:** This Agreement is governed by the laws of the Province of Alberta, Canada. Each of the parties hereto irrevocably attorns to the jurisdiction of the courts of the Province of Alberta.

10. **Customer Support:** For Software UPDATES and UPGRADES, and regular customer support, contact the NovAtel GPS Hotline at 1-800-NOVATEL (U.S. or Canada only), or 403-295-4900, Fax 403-295-4901, e-mail to wpsupport@novatel.ca, website: http://www.novatel.com or write to:

NovAtel Inc.
Waypoint Products Group
1120 - 68 Avenue NE,
Calgary, Alberta, Canada T2E 8S5

Warranty

NovAtel Inc. warrants that during the warranty period (a) its products will be free from defects and conform to NovAtel specifications; and (b) the software will be free from error which materially affect performance, subject to the conditions set forth below, for the following periods of time:

Computer Discs
Ninety (90) Days from date of sale

Software Warranty
One (1) Year from date of sale

Date of sale shall mean the date of the invoice to the original customer for the product.

Purchaser's exclusive remedy for a claim under this warranty shall be limited to the repair or replacement at NovAtel's option and at NovAtel's facility, of defective or nonconforming materials, parts or components or in the case of software, provision of a software revision for implementation by the Buyer.

All material returned under warranty shall be returned to NovAtel prepaid by the Buyer and returned to the Buyer, prepaid by NovAtel. The foregoing warranties do not extend to (i) nonconformities, defects or errors in the Products due to accident, abuse, misuse or negligent use of the Products or use in other than a normal and customary manner, environmental conditions not conforming to NovAtel's specifications, or failure to follow prescribed installation, operating and maintenance procedures, (ii) defects, errors or nonconformities in the Products due to modifications, alterations, additions or changes not made in accordance with NovAtel's specifications or authorized by NovAtel, (iii) normal wear and tear, (iv) damage caused by force of nature or act of any third person, (v) shipping damage, (vi) service or repair of Product by the Purchaser without prior written consent from NovAtel, (vii) Products designated by NovAtel as beta site test samples, experimental, developmental, preproduction, sample, incomplete or out of specification Products, (viii) returned Products if the original identification marks have been removed or altered or (ix) Services or research activities.

THE WARRANTIES AND REMEDIES ARE EXCLUSIVE AND ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, WRITTEN OR ORAL, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE ARE EXCLUDED. NOVATEL SHALL NOT BE LIABLE FOR ANY LOSS, DAMAGE, EXPENSE, OR INJURY ARISING DIRECTLY OR INDIRECTLY OUT OF THE PURCHASE, INSTALLATION, OPERATION, USE OR LICENSING OR PRODUCTS OR SERVICES. IN NO EVENT SHALL NOVATEL BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE DUE TO ANY CAUSE.

Chapter 1   Introduction and Installation

1.1   RTKNav Overview

This chapter gives instructions on how to install RTKNav and includes descriptions of the various components of RTKNav.

1.1.1   RTKNav

RTKNav is a real-time PC software program that uses GPS and GLONASS satellite observations to compute position and velocity. The software accepts pseudorange and carrier phase data from a range of different receiver types and computes sub-meter, sub-decimeter or centimeter positions depending on the conditions of data collection, receivers used and application dynamics. RTKNav can be used in forward RTK or inverse RTK applications and can process up to 20 baselines simultaneously.

RTKNav applications are in the following two categories:

1. **Forward RTK processing**: The base station transmits raw data through a radio link to the rover, where RTKNav is running. This is the master input stream. At the rover, a GPS receiver is connected directly to RTKNav. This is the remote. Normally, RTKNav is used to navigate or track the location of a vessel, vehicle or aircraft.

2. **Multi-remote Inverse RTK**: RTKNav runs at the base station and connects directly to the master receiver. 1 to 20 mobile units are continuously transmitting raw data back to the base for processing. The base can be moving or static. A moving base station can either be attached to the same body as the rover, so that a heading can be computed, or attached to a different moving body where a vector and azimuth can be computed. If the base station is attached to the same body a measured baseline length constraint can be used. RTKNav plots the location of each of the mobile units. Their solution can also be sent to another application.

Currently, RTKNav supports the GPS receiver inputs that are listed in the shaded box.

GPS receiver inputs currently supported by RTKNav

- Magellan/Ashtech AC12/G12/DG16/ MACM/Z12/Z12DBEN/MACM_GSU1/ MACM_GSU2
- Canadian Marconi (NovAtel) AllStar/ SuperStar
- Javad GRIL/OEM
- Navcom NCT
- NovAtel OEM 3/OEM4/OEMV
- Trimble 4x00/SSx
- Ublox Anataris
- CSI CLSX

RTKNav offers a full-featured Windows interface. This interface comes with a Configuration Wizard, plots up to 20 remotes, and shows the tracking status of each connected receiver. The red-yellow-green light means that system operation can be checked by quick inspection. For moving baseline applications, a bull's-eye view is possible, along with range and bearing display. Waypoint navigation is also supported.

For more information on RTKNav, please see *Chapter 2* on *Page 17*.

1.1.2   MultiEngine

MultiEngine has the same basic functionality as RTKNav but with a simplified user interface. In addition, MultiEngine can process asynchronous rover data with long delays of up to several minutes.

For more information on MultiEngine, please see *Chapter 5* on *Page 60*.

1.2   Utilities

This section lists and describes the utilities included with the software.

1.2.1   Concatenate, Slice and Resample

This utility edits raw data files produced by RTKNav/WLOG. If you choose to log in the raw binary format of the GPS receiver, the files have to be converted to Waypoint's GPB format before this utility can be used. See *Chapter 3* on *Page 35* for more information.

1.2.2   GPB to Rinex Converter

This utility produces a RINEX file from GPB files. See *Chapter 3* on *Page 35* for more information.

1.2.3   GPB Viewer

This utility allows you to view the raw binary data collected by the GPS receiver in order to detect any problems. Measurement values and position records are among the fields viewed here. See *Chapter 3* on *Page 35* for more information.

1.2.4   GPS Data Logger

This utility facilitates GPS data logging directly from a variety of GPS receivers under a Windows 2000, XP and Vista environment. Tagging stations and satellite lock plots are also included. Logging can be performed directly into Waypoint's custom format. See *Table 1* for supported receivers. For more information see *Chapter 3* on *Page 35*.

1.2.5   Hardlock Upgrade Utility

This utility reads the information on your Sentinel key and allows you to enter upgrade codes from Waypoint's customer service group.

1.3   Installation

1.3.1   What you Need to Start

- At least two suitable GPS receivers are also required. One is the base and the other ones act as remotes. The antennas should have an open view of the sky in order to work with RTKNav. Placing them in a window may not provide enough satellites for navigation.

- A serial port is required if you wish to output serial records to another device.

✉   This can also be performed via TCP/IP (TCP mode only).

Table 1: Receivers Supported by Data Logger

Make	Model	Data Logging Windows
NovAtel	OEMV OEM4 OEM3 OEM2 RTCM3	✓ ✓ ✓ ✓ ✓
CMC/ NovAtel	Superstar II Allstar	✓ ✓
CSI	DGPS MAX	✓
Javad	All models	✓
NAVCOM	OEM GPS	✓
Magellan	Real-Time	✓
Trimble	4000 series (RT) 5700 SSx	✓ ✓ ✓
U-blox	Antaris	✓

You also need a hardlock key connected to the computer parallel or USB port.

1.4 Definitions and concepts

This section describes some of the terms relating to serial and network communication ports.

✉   In the following discussion, the term Win32 denotes Win2000, Windows XP and Windows Vista. RTKNav has been tested on all of these operating systems.

Assigned Port Number
RTKNav, and RtEngine use an assigned port number to reference a serial or network port. Assigned port numbers range from 1 – 1023, and they have no correlation to serial COM ports or network port numbers. Assignment is usually sequential.

COM port (computer)
The Windows system defines a number of serial communications ports, depending on what serial devices are installed. Generally, you will have COM1 and COM2. Your mouse may use one of these. Serial boards or Serial - USB devices are shown as COM ports in the Win32 system. RTKNav reads the Win32 system and determines which COM ports you have and reads measurement data from your GPS receivers. Ports are usually defined between COM 1 – COM 20.

COM port (GPS receivers)
Currently, most GPS receivers have one or more serial COM ports. RTKNav can communicate from a computer COM port to either GPS receiver port A or B.

✉   RTKNav requires two-way communication with receivers to send configuration commands and to receive data.

Baud Rate
RTKNav asks you to choose the data rate to communicate over your serial port. Baud rates from 4800 bits per second to 115200 bits per second are supported. RTKNav runs reliably up to 115200 bps (baud). Your GPS receiver must support the same baud rate as your computer. RTKNav polls the receiver to set its baud rate to the correct setting, providing the receiver supports that rate.

Network Port
RTKNav sends or reads GPS data that has been broadcast over a network. The term "*network port*" describes a communication port through which a data stream is being sent over a local or wide-area network.

Network Port Number
Like COM ports, network ports are assigned numbers so that each data stream is identified with an associated integer number. Network port numbers from 1 to 1024 are generally reserved for use by the operating system, FTP, Internet and so on. The maximum port number allowed is 65356. RTKNav defaults to port numbers starting at 6001. RTKNav attempts to assign different port numbers for you. You are free to type in your own port numbers. These port numbers uniquely identify each GPS binary data stream or each output ASCII data stream from RTKNav to other network users.

IP Address
Each computer on a local or wide area network is given a unique set of 4 numbers which identify this computer to all other computers on the network. Local network computers (nodes) will tend to have IP addresses such as 192.xxx.xxx.xxx. Computers that have static IP addresses are capable of receiving data over the Internet. RTKNav uses these addresses in TCP mode to send data anywhere.

Network Protocol
RTKNav supports 3 types of network data protocol. Your choice of data protocol depends on the operating system that you have and whether you wish to send the data over your local network or over the Internet to a remote location. Local networks is the network in your office or, for example, on your ship. Wide area networks communicate all over the world on Internet connections. The 3 types of network protocols supported are listed in the shaded box.

Re-Broadcast Data
This protocol collects GPS raw binary data from a serial port and sends this data over a local or wide area network to another window on your computer, another RTKNav user in your local network (UDP or MULTICAST protocol) or another RTKNav user in some other part of the world (TCP protocol). This allows multiple instances of RTKNav to process the same data from the same group of GPS receivers. This is useful if you want one copy of RTKNav to process data from your entire group of GPS receivers, while another copy only examines data from a subset of your receiver group.

The network protocols that are supported:

1. UDP (User Datagram Protocol)
UDP is used strictly on local networks. UDP is good for sending data to your computer or another computer in your office. If a data stream is sent on UDP, then it is sent to every computer in the local network. You must receive this data stream in UDP and be on the same port number as the sender. For instance, if RTKNav is asked to re-broadcast a serial data stream from a NovAtel GPS receiver by UDP on port 5001, then every computer on the local network will receive UDP data on port 5001. The receiver must specify UDP, port 5001 to get the data into another instance of RTKNav running somewhere else on the network.

⊠ In Win32 operating systems, UDP is always assigned an IP address of 255.255.255.255. This may not be true for UNIX-based systems. UDP should work on Windows 2000, XP, Vista, and NT.

2. Multicast protocol
Like UDP, Multicast is a good choice for local networks.

⊠ All protocols, including Multicast will "loop" network data to your own computer. This means that you can use network ports to send data to another instance of RTKNav on your own computer.

Multicast, like UDP, sends data to every computer in a local network. The difference between UDP and Multicast is that you must type in an IP address. This address denotes a user group. All Multicast Win32 users must join a group with IP addresses between 224.0.0.0 and 240.0.0.0. RTKNav defaults to a user group address of 234.5.6.7.

Multicast users must type in a group IP address, whether they are sending or receiving data.

⊠ Multicast may not work on some Win95 computers. UDP may be used in an equivalent fashion.

3. TCP
TCP is a point to point communication protocol. If you know the static IP address of a remote user (and they have opened their firewall), you can send GPS data anywhere in the world.
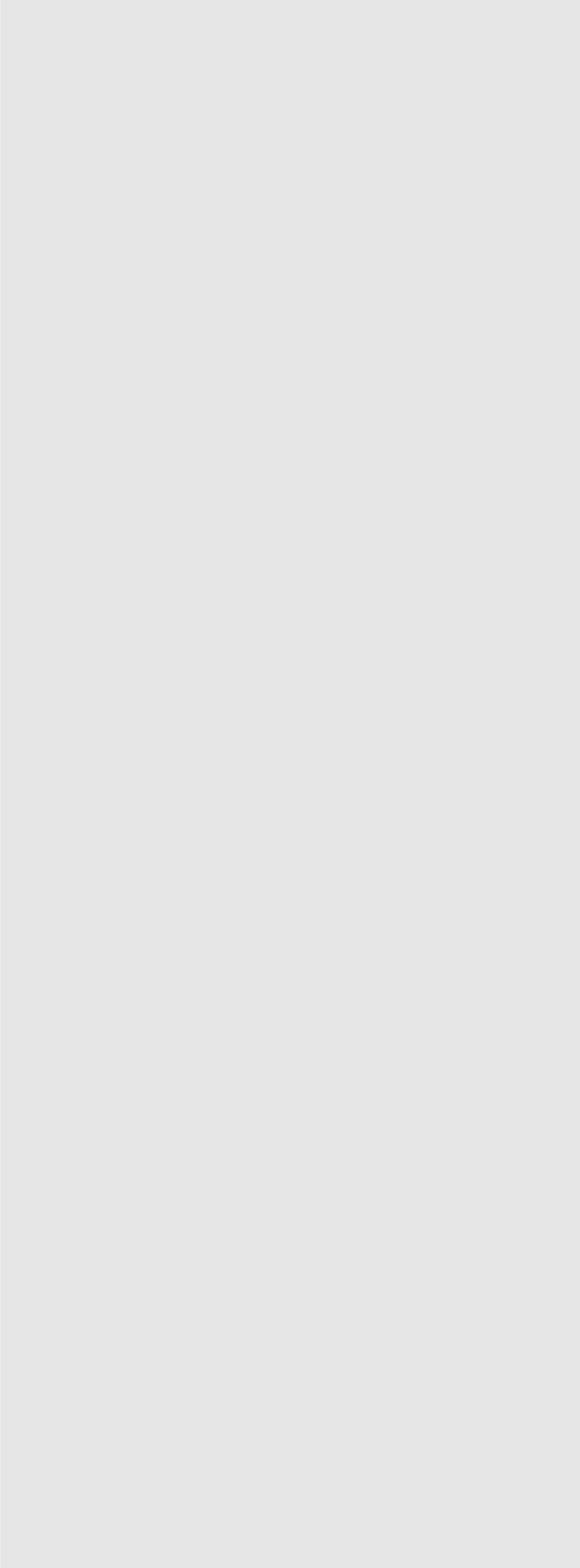
You might also find it useful to have another remote computer store your raw binary measurements for future post-processing.

✉   RTKNav data received over a network can also be re-broadcasted to other network ports.

2.1 RTKNav Overview

RTKNav is a Windows-based real-time GPS processing program with a graphical user interface (GUI) that is easy to navigate. RTKNav supports one base station receiver and one or more remote station receivers.

RTKNav collects real-time GPS measurement data from either serial or network ports. Binary measurement information is re-directed (re-broadcasted) to local or wide-area network ports and allows other instances of RTKNav to process the same data at another location. The RTKNav R3 license only allows one instance of the software to be run at a time.

The processed data information can be output in a number of ASCII formats to either serial or networks ports. This permits other programs to receive processed high precision coordinate and quality information either locally or by the Internet.

2.2 Start a Project with RTKNav

The following section describes step-by-step how to start RTKNav projects.

2.2.1 Run RTKNav

The steps to run RTKNav are in the shaded box.
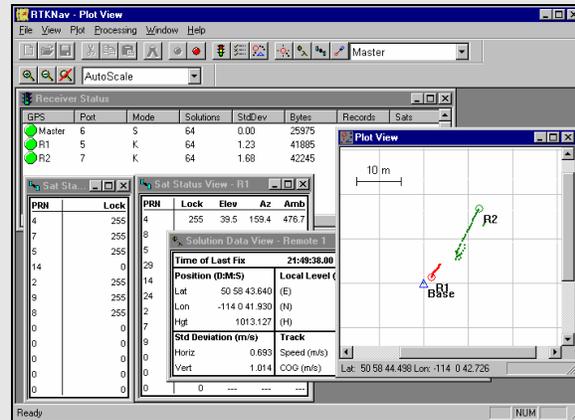
How to run RTKNav

1. Go to the Windows Start menu.

2. Choose either RTKNav with a maximum of 3 receivers (RTKNavR3) or RTKNav with a maximum of 20 receivers (RTKNavR20).

⊠ RTKNavR20 takes up considerably more memory and CPU then RTKNavR1 or RTKNavR3. RTKNavR20 should only be used if you have more than 3 remote GPS receivers.

How to create a new project

1. Select *File | New Project*.

2. Navigate to where you would like to save the project.

3. Type a filename for your project and press *Save*. The *Configure Project* window appears for adding receivers to the project with the *Configure Project* window
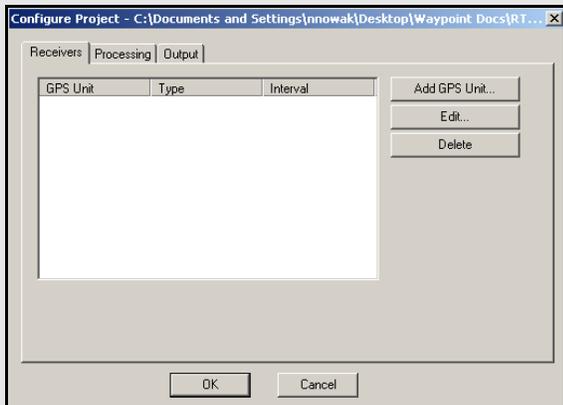
4. Define the communication parameters.

5. Click on *Add GPS Unit*. Use this button to add the base receiver and all of the remote receivers to the project.

6. Click *Next*.

7. Select GPS type and, if required, add coordinates and height.

8. Click *Next*. The GPS Mode window appears.
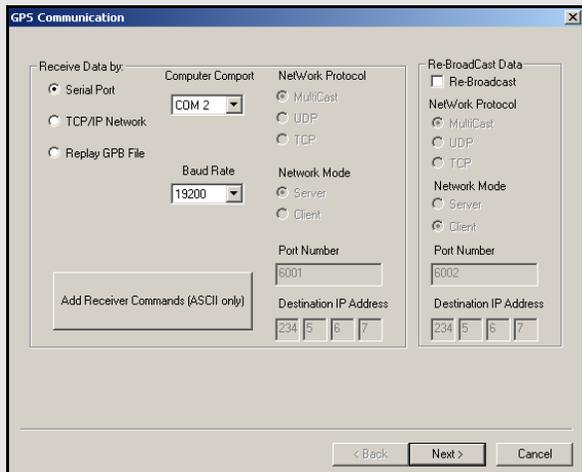
Continued on the next page.

2.3 File Menu

2.3.1 New Project

To start processing data for the first time, create a new project. When you create a new project, RTKNav forms two project files, an IN (Input) file and a CFG (Configuration) file. The IN file contains information about communication ports. The processing engine RtGNSSx.dll uses the CFG file to perform GPS processing functions.

The steps for creating a new project are in the shaded box.

2.3.2 Add GPS Receiver

GPS Communication

RTKNav sends processed coordinate and quality information for each baseline out either a serial or network port. RTKNav also allows for multiple output ports.

Receive Data by:

Serial Port
This option lets you choose a COM port that Windows has available on your system.

Computer Comport
RTKNav recognizes any COM port that Windows has available on your system.

Baud Rate
Baud rates of up to 115200 are allowed and are reliable.

TCP/IP Network,
Use this option to receive GPS measurement data by network.

For more information on network protocol, network mode, or port numbers, see *Definitions and concepts* on *Page 14*.

Destination IP Address
To send data by network port, you must specify the IP address of the output destination. In the case of MultiCast, you must join a MultiCast group all on the same IP address defined between 224.0.0.0 and 240.0.0.0. UDP users on Win32 machines have to use IP address 255.255.255.255. TCP users must know the IP address of their target computer, whether local or wide area. For wide area networks, this is typically a static IP address.

Re-Broadcast Data
Re-broadcast your serial or network data to your own computer or to a computer anywhere on your network. This is very useful if you need another user to simultaneously process the same data that you collect. This user does not have to be hooked directly to the GPS receivers. Only the original RTKNav has to be connected directly.

This may be useful, if you want one copy of RTKNav to process data from your entire group of GPS receivers, while another copy only examines data from a subset of your receiver group. You might also find this useful, if you simply wish to have some other remote computer store your raw binary measurement for future post-processing.

> ⊠ RTKNav instances receiving data over a network can also re-broadcast this data to other network ports.
>
> You cannot re-broadcast data when replaying GPB files.

Replay GPB File
If you have already collected GPS data in GPB format, you can replay it through the RTKNav processor to simulate the real-time processing. See *Section 2.7,* on *Page 29*.

Select GPS Type
This window allows you to select your receiver type, designate your receiver as a base or remote and enter base station positions

How to create a new project (continued)


9. Select the port you will use, the interval to output data and the processing mode. If you want to log data, select the file type and the file prefix.

10. Repeat steps 5 - 9 to add more receivers.

11. Click *OK* once all the receivers for the project are loaded. This creates your new project.

> ⊠ Users of NovAtel and Ashtech receivers have the option to send ASCII commands to the receiver. This feature can be used to either disable commands or over-ride various receiver commands. Refer to the receiver manual for information on such commands.

RtGNSSx.dll Processing Capabilities

- **Float Code/Phase** – no ambiguity determination; accuracy: sub-meter to decimeter

- **On-the-Fly Ambiguity Determination** – Advance Real Time Kinematic (ARTK) with code/phase measurements; accuracy: centimeter level. This can be invoked for either single or dual frequency measurements
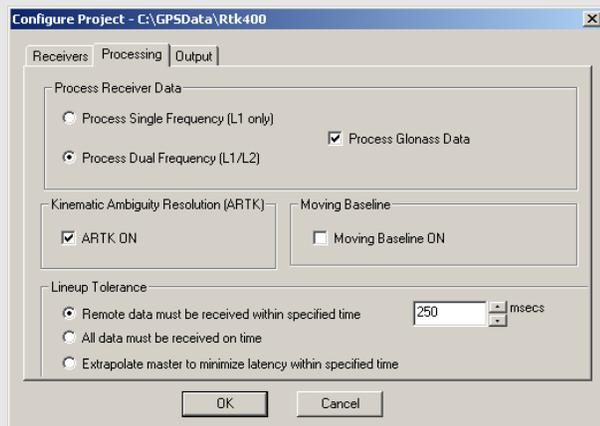
If the starting remote station coordinates are known, you can perform a remote initialization here. This significantly increases the speed at which centimeter level positioning can be performed, and is especially important if you engage for deformation monitoring applications.

Orthometric heights can also be entered if there is complete geoid coverage of the area of interest. See *Section 2.5.3.6,* on *Page 28* for more information about geoids in RTKNav.

GPS Mode

This window establishes whether the data is kinematic or static, whether you intend to log this data to hard disk and at what data rate. All of this information is important in configuring communication to the GPS receiver. You can also use this window to configure either the primary or secondary GPS receiver comports.

Which Port on the GPS Unit will You be Using?
Some receivers have primary and secondary comports: A, B, C, or D.

At What Interval Would You Like the GPS Unit to Output Data?
Data intervals can be as high as you wish. This allows you to type a custom output data interval. Typical data intervals for real-time processing are from 1 to 10 Hz (Hertz).

In What Processing Mode Will This Unit Start?
It is important to define whether the data is static or kinematic in nature. Most applications have static base stations but for moving baseline mode, define the base station data as kinematic. For most applications, the remote receivers is kinematic. This is the default setting.

Log This Unit's Data To A File?
To log data to disk, store it in either Waypoint's GPB binary measurement format, or in a byte by byte as a raw file. The *GPB File* or *Raw File* modes are useful if you plan to post-process the data for later analysis.

2.3.3   Processing

The processing engine that provides RTKNav with coordinate information is called RtGNSSx.dll. RtGNSSx.dll is capable of processing in the ways listed in the shaded box.

The default processing mode for RTKNav is single frequency float code/phase, with the base station being considered static and the remotes being considered kinematic.

Change these default settings if you wish to achieve any of the following:

- Single frequency processing

- Disable on-the-fly Advance Real Time Kinematic

- Change the time allowed for which data can be recieved from all remotes before processing is performed

- Moving baseline processing and/or heading determination

Receiver Data

Dual frequency measurements help ARTK considerably, and permit the minimization of ionospheric effects. Therefore, if dual frequency receivers were used, it is beneficial to ensure this radio button is selected.

Advance Real Time Kinematic (ARTK)

Check this box to resolve phase ambiguities on the fly. This is very important for if you require centimeter accuracy. This can be done for single or dual frequency receivers. Baselines should be as short as possible for optimal reliability; less than 3 km for single frequency and less than 30 km for dual frequency. More related options are available under *Processing | Options.*

Line-Up Tolerance

This setting has to do with the availability of the serial or network data coming from the remote GPS receivers.

Remote data must be received within specified time

This is the default and contains 250 milliseconds. This means that RTKNav waits up to 250 msec to see if all the baselines are sending data. If any baselines fail to send their measurements in this time frame, RTKNav computes positions for the baselines that did send data, and ignore the baselines for the remote receivers that failed to send data.

> ✉ For computing at 10Hz, this value must be 100 msec at most. For long-term projects, the default option should be selected.
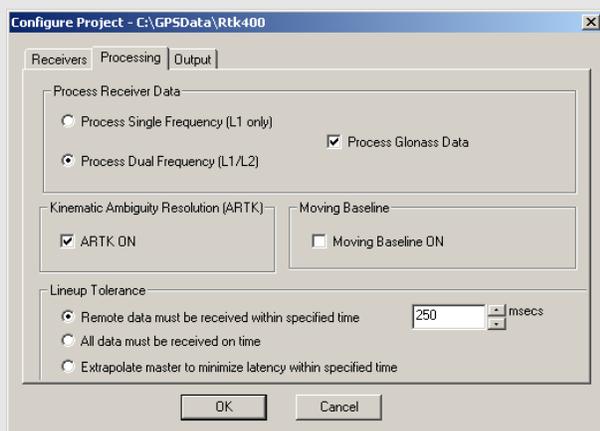
Applications that might require additional processing commands are:

Moving Baseline applications – any ship to buoy(s) or airplane-to-airplane(s) applications require the moving baseline option to be turned ON. In addition, moving baseline users should define their base station data as kinematic, not static.

Orthometric heights can be computed real-time using a Waypoint Geoid file (WPG) using the geoid32.dll. You can enter the base station coordinates in ellipsoidal height or orthometric height and have the output as ellipsoidal height or orthometric height.

✉ Although the RtGNSSx.dll Kalman filter can be commanded with a large number of options, most applications are able to use the default options as set internally by RTKNav.

Moving baseline users are an important exception to this, as RTKNav must be told that you intend to use it in Moving Baseline mode. Remember that once an IN and a CFG file are created and adopts all values in these files upon opening an existing project.

All data must be received on time
The option has a major disadvantage if you are in a project and one GPS receiver fails to send data. In this case, RTKNav does not compute data for any baselines. If you expect some remote GPS receiver to go off-line for any reason, then use the default option.

Extrapolate master to minimize latency within specified time
For projects where the base station data arrives either very latent or at a lower interval, the setting may be used. In such cases, the master station data is extrapolated using time history information. For this option, the user defined option EXTRAPOL_EPOCHS may need to be added.

Moving Baseline
Moving Baseline ON
Ensure this is checked if you have a moving baseline application such as a ship-to-buoy(s) or airplane-to-airplane(s). In addition, moving baseline users should ensure that they have define their base station data as kinematic, not static.

The GPS processing engine RtGNSSx.dll can be configured with a wide variety of commands. Some of these are very useful or even essential for some applications. Some applications that might require additional processing commands are listed in the shaded box.

2.3.4   Output

You can choose to send the output ASCII data to serial or network ports, or to write to disk. If you choose write to disk, your output file is saved in your working directory and is given the same name as your project, with an optional file prefix.

See *Section 2.3.2,* on *Page 18* for more information.

Add Receiver Commands (ASCII only)
To add a selection of ASCII records for output, select a record type. Before clicking the *Add* button, select the interval to output this record. Output ports can be used to send ASCII records out a serial or Ethernet port. See *Chapter 4, on Page 51* for more information.

2.3.5   Open Project

If you already have a project created or a defined IN file then use this to open the project.

2.3.6   Close Project

Closes the project.

2.3.7   Save

RTKNav saves the project. All selected options are also saved.

2.3.8   Save As

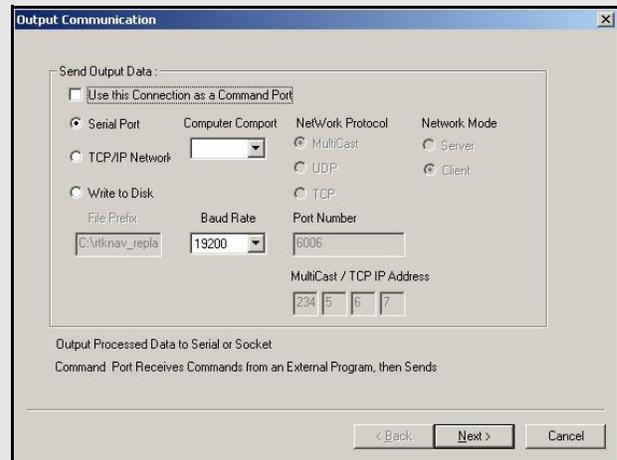Use this to save the project under a new name. This will not overwrite the original project.
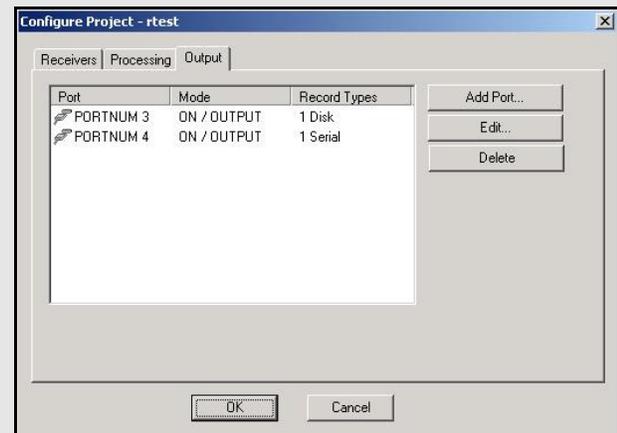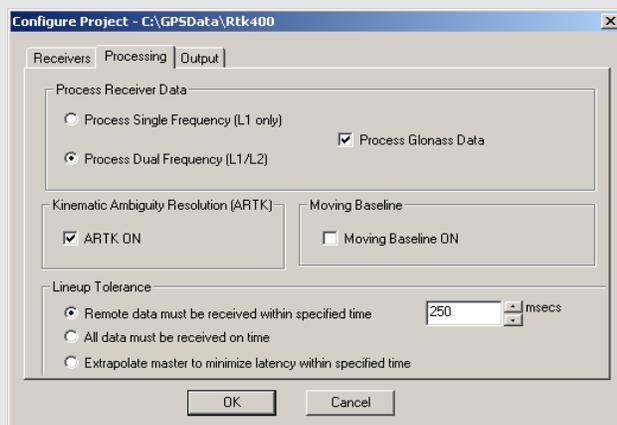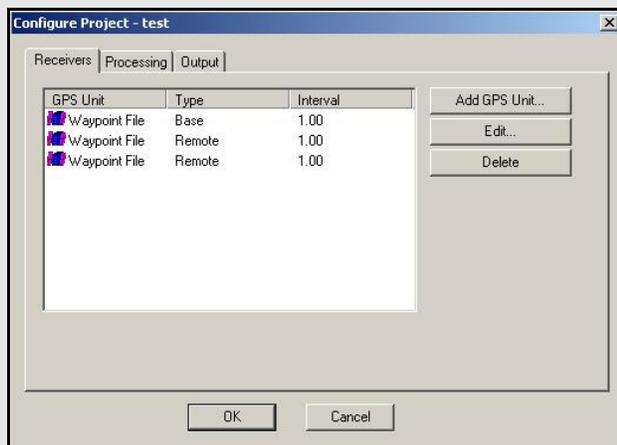
2.3.9   Log Status Data

Provides the status of the software. For example, if start-up was successful or a receiver drops out.

2.3.10   Exit

Exits the program.

2.4 View Menu

2.4.1 Project Configuration

Receiver

Add GPS Unit
Accesses the projects configuration window. See *Section 2.3.2,* on *Page 18* for more information.

Edit
Opens the *Edit GPS Unit Wizard* window. Select the unit and click *edit* to access the edit options.

Delete
To delete a unit, select the unit and press delete.

Processing
Accesses the projects processing configuration window. See *Section 2.3.3,* on *Page 20* for more information on the settings in these tabs.

Output
Accesses the projects output configuration window. See *Section 2.3.4,* on *Page 23* for more information on the settings in these tabs.

2.4.2 Launch/Stop Google Earth

This option allows users with Google Earth installed to view the positions of all their remotes. See the print screen on the next page. Each remote is plotted with icons that are color-coded according to the epoch's computed quality number. The icons are also directional, based on the computed course-over-ground of the remote. Real-time plotting in Google Earth is done by creating a network link that continuously reads the KML file written by RTKNav. Note that the maximum refresh rate in Google Earth is one second.

2.4.2.1 Show Breadcrumb Trail in Image

Enabling this option will force Google Earth to display previously-computed solutions for each remote. This gives you the ability to view the solution history. If you are only interested in the most recent solution, you can disable this option. In this case, only one icon will be displayed for each remote.

Number of crumbs to display: Specifies how many previous solutions to display for each remote. Raising this value too high can lead to system performance degradation and can also make the map look crowded, depending on the number of remotes in the project.

Refresh Breadcrumbs Every n Seconds: Controls the rate at which the bread crumbs are saved to the KML file. This, in turns, controls how often Google Earth updates the breadcrumb trail in the display. It is not recommended that this value be lowered, as it will force Google Earth to use more resources, which can slow the system down.

⊠ Occasionally, Google Earth's network link function may fail due to a conflict between its attempt to read a KML file that RTKNav has open for writing. The auto-updating can be re-started by right-clicking on the network link in Google Earth (found on the sidebar as *Auto Update)* and selecting *Refresh*.

2.5 Processing Menu

2.5.1 Start Processing

Assuming all of your serial and network connections have been made properly, RTKNav starts processing when you select *Processing | Start Processing* or click on the icon with the green circle.

⊠ Prior to processing, the receiver icons are RED to indicate no data is being logged.

When processing begins, RTKNav sends configuration commands to each GPS receiver at the baud rates you have chosen**.** If the GPS receivers respond to these configuration messages, the icons turn YELLOW.

If RTKNav receives measurement data from all of the receivers and ephemeris data from at least one receiver, baseline solutions are computed. If a baseline solution has been successfully computed, the icons turn GREEN
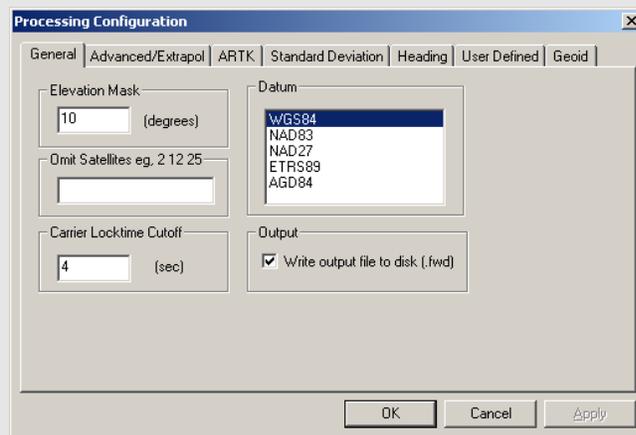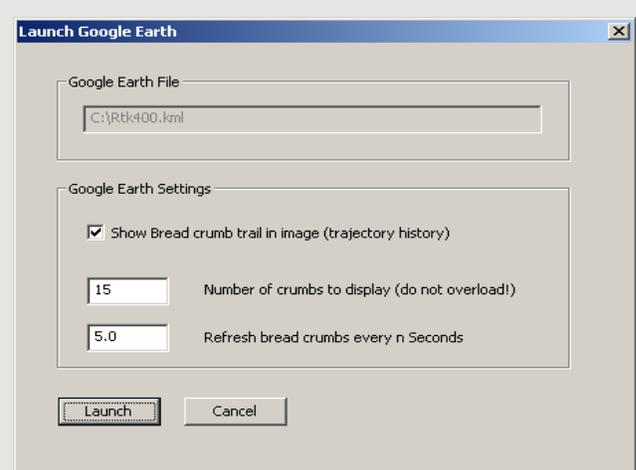
2.5.2 Stop Processing
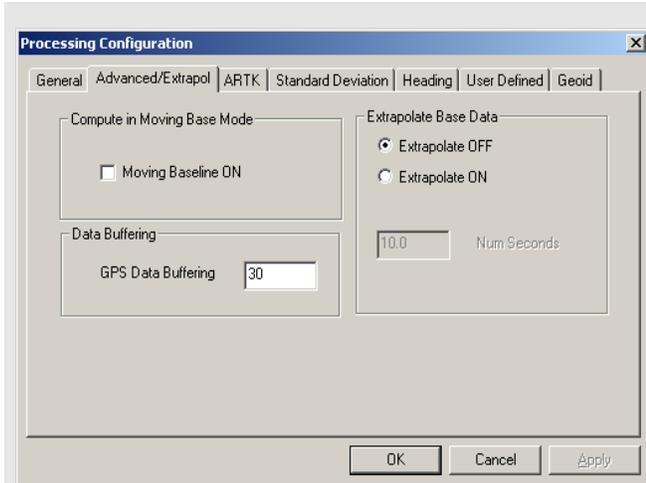
Use this option to stop processing.

2.5.3 Options

2.5.3.1 General

Elevation Mask
Satellites below this mask angle are ignored. A 10-degree elevation mask is suitable for most kinematic

applications. However, as baseline lengths increase, a 15- degree mask becomes more suitable for monitoring applications.

Omit Satellites
This is a list of satellite PRN numbers to ignore. Multiple satellite PRNs must be separated by spaces. This option is only useful in post-processing for eliminating problematic satellites.

Datum
By default, the processing datum is WGS84. This processing datum also works well if base station coordinates are entered in NAD83. Otherwise, it is important to enter the base station coordinates in the selected datum.

Write Output File to Disk
Writes an ASCII solution file to hard disk. One file is written for each remote. This file contains information about every epoch of data processed for each baseline. A more detailed explanation of this is given in *Chapter 4* on *Page 51*.

Carrier Locktime Cutoff
This value indicates the number of seconds of continuous carrier lock before the receiver's phase measurements are actually used by RTKNav. Most receivers, even those of reputable high quality, put out unreliable phase measurements for the first few seconds after acquiring lock.

2.5.3.2 Advanced/Extrapol

Moving Base
Check this *ON* for moving baseline mode. If this box was selected previously in the Configure Project window, it is already checked here.

GPS Data Buffers
This is the number of epochs of data that is searched in order to line up base and remote receiver measurement epochs. It can normally be ignored. For most applications, 30 epochs is sufficient. However, if data rates and latencies are high, this number may need to be increased. For users replaying GPB files, this number must be increased, depending on how much data is being replayed. If the replayed data stops unexpectedly, the user must increase this number to allocate more buffers to be read.

Extrapolate Base Station Data

Enabling base station data extrapolation allows the software to generate (extrapolate) its own base station measurement data when no base station exists. This can be done to minimize

the bandwidth usage of the radio/network. For example, the rover could be computing solutions at 10Hz and the base station could be transmitting data only every 5 seconds. In such a case, base station data is only transmitted once for every 50 solutions computed. Base station data extrapolation can also be done to safeguard against radio/network latencies or occasional lost base station records. For example, if the rover is receiving the base station data with a 400ms delay, if extrapolation is disabled, then the solution will be delayed by at least this amount of time. However, if extrapolation is enabled, as soon as the rover measurement is received, a position can be computed. Also, if a base station record is dropped by the radio/network, the rover can fill this in with the extrapolator. Extrapolation should only be done in forward RTK. The extrapolation adds approximately 5 -7 mm or error per second of extrapolation from the last base station record received. The maximum extrapolation threshold should be set according to the application's requirements. It is not recommended to ever exceed 30 seconds. Extrapolation can not be done in moving base applications as the base station must be static. When using extrapolation, it is important to use the correct base station coordinates (within a meter).

2.5.3.3   ARTK

Kinematic Ambiguity Resolution
For high-accuracy applications, this option needs to be enabled to ensure that ARTK is used for integer ambiguity resolution. Leaving this setting disabled will result in float solutions.

Enable Single Frequency ARTK
L1-only users can enable ARTK via this setting. It should be noted, however, that ambiguity resolution tends to be much less reliable on single frequency data. This setting is only recommended for short baselines in open-sky conditions and/or moving-baseline applications.

ARTK Quality Level
This value determines how strict ARTK is in signaling a fixed integer solution as a pass. The higher the Q-level, the less likely that ARTK will accept a fix. For dual frequency data, it should be left at Q1, but if ARTK is computing many incorrect fixes, it may need to be raised. If ARTK is enabled for single frequency, then this value is automatically set to Q4 internally.
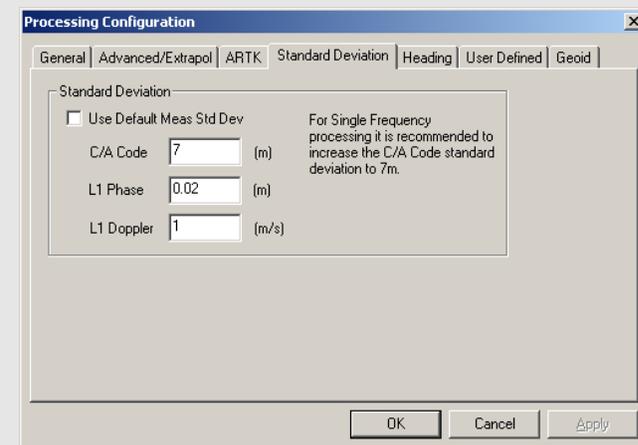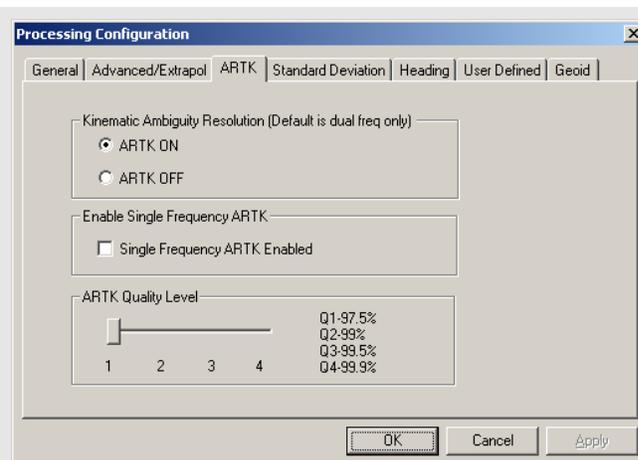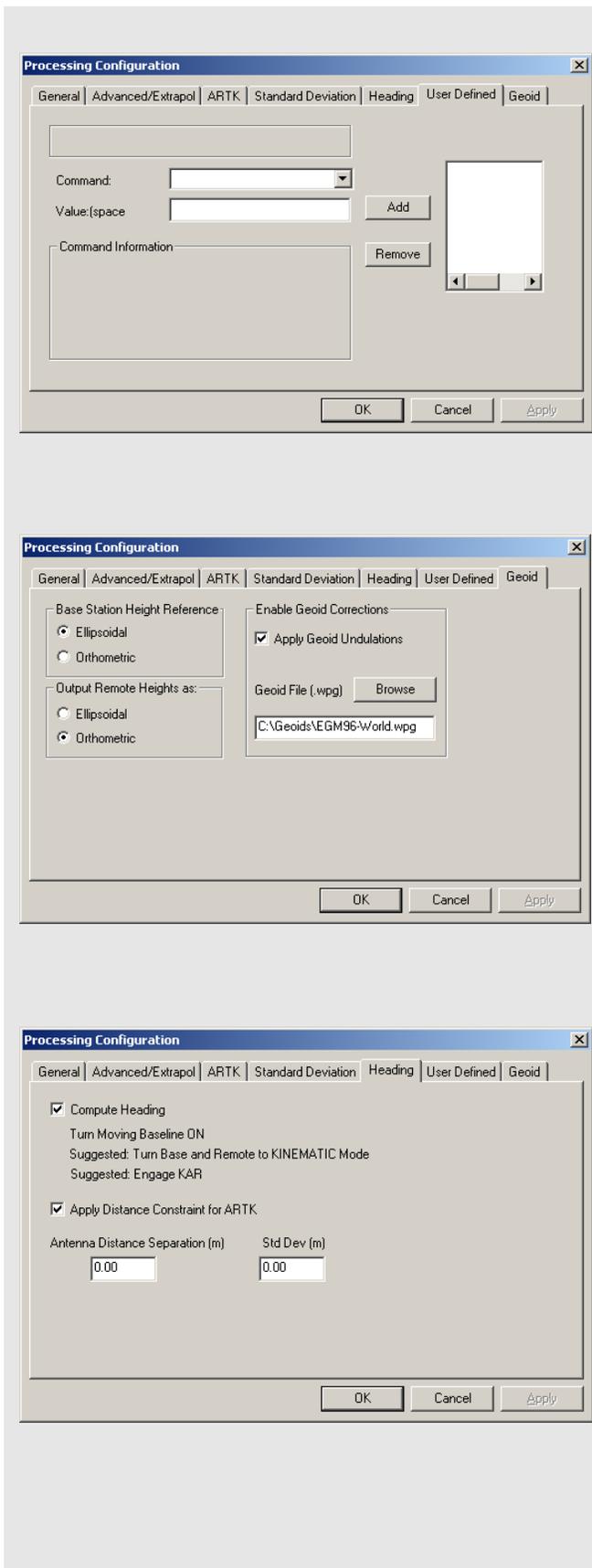
2.5.3.4   Standard Deviation

These indicate the a priori standard deviations applied that give weights to the various measurements used in the Kalman filter. The rejection tolerance values determine when the filter should reset itself if it is not able to fix measurements that have post-filtering errors above the given tolerances.

Using realistic values for the code and carrier standard deviations improves float solution convergence. For instance, many narrow correlator receivers have a C/A code measurement of 1 to 3 meters, this is much lower than the default value of 7 meters. Normally, you do not have to change these values. In extreme GPS environments, it may be useful to give more weight to the code measurements and less weight to the phase measurements. Tracking sounding rockets is an example of this. These carry single frequency receivers to heights of 1000 km. Changing the carrier phase measurement deweights the carrier phase measurement and therefore dampens the effect of uncorrected atmosphere to 0.20 m and the code to 2.0 m.

Use Default Meas Std Dev

Recommended for dual frequency data. Use values of 2.25 m, 0.02 m, and 1 m/s measurements, respectively.

2.5.3.5   User Defined

There are numerous commands that can be sent to RGNSSx.dll. The *User Defined* options are all listed in the attached Appendix. These are generally used to modify commands that are not normally used.

2.5.3.6   Geoid

This feature computes orthometric heights while processing real-time or when replaying a GPB file. It is important that you have a valid WPG file that spans the area of interest so that the geoid height can be computed.

Base Station Height Reference
You can select whether your base station coordinates entered were ellipsoidal height or orthometric height.

Output Remote Height
You can also specify the output for all the remotes as either ellipsoidal height or orthometric height.

Enable Geoid Corrections

Enabling this option prompts RTKNav to calculate orthometric heights
when processing. Click *Browse* to locate the path of the WPG file that RTKNav uses as a source of geoidal undulations. The file must span the area of interest. Otherwise, RTKNav returns an error.

2.5.3.7 Heading

Compute Heading
This option must be enabled for moving baseline applications requiring heading computations. For best results, ARTK should be enabled. Even with ambiguities resolved, heading accuracy will be dependent on the distance between the moving antennas.

Apply Distance Constraint for ARTK
Enabling this option will help reduce the amount of time needed for ARTK to resolve ambiguities, particularly for L1-only applications.

Antenna Distance Separation: This value represents the distance, in meters, between the L1 phase centers of the moving antennas.

Std Dev: This value represents your confidence in the specified distance. It is important that it be set to a realistic value because it is used to engage ARTK if the computed distance varies too much from the specified distance. Setting this too low can cause continuous ARTK re-engagements.

2.6   Help

2.6.1 About RTKNav

Provides version number information as well as memory and disk space availability.

2.7   Replaying Data in RTKNav

The steps on how to replay data in RTKNav are in the shaded box. Some notes about replaying data in RTKNav include the following:

- The GPB files must overlap so that the base and remotes can be processed differentially. RTKNav can handle data gaps. If necessary, use the Concatenate, Splice and Resample utility. See *Concatenate, Splice and Resample Overview* on *Page 40*.

How to replay data in RTKNav

1. Click *File/New Project* to start a new RTKNav project.

2. Click *Add GPS Unit.*

3. Select *Replay GPB File* and click *Next.*

4. Identify the unit as either the master or the remote file in the *Select GPS Type.*

5. Enter the base station coordinates. If it is a base station, specify the elevation value as ellipsoidal or orthometric. Click *Next.*

6. Locate the path of the GPB file and an appropriate ephemeris (EPP file) to go along with it in the *GPS Mode* window.

7. Specify whether this data is static or kinematic for the added receiver. Check the mode in the *GPB Viewer (Section 3.2,* on *Page 35*) and make sure that the GPB file has the appropriate static/kinematic mode. Click *Finish.*

8. Follow steps 2-7 to add other GPB files to the project.

9. Select the *Processing* tab and configure the processing options for the project. For more information on the processing options, see *Section 2.3.3,* on *Page 20.*

10. Select the *Output* tab and define different methods/formats of output. For more information, see *Section 2.3.4,* on *Page 23.* The number of data buffers must be increased when replaying data, otherwise the project stops processing prematurely.
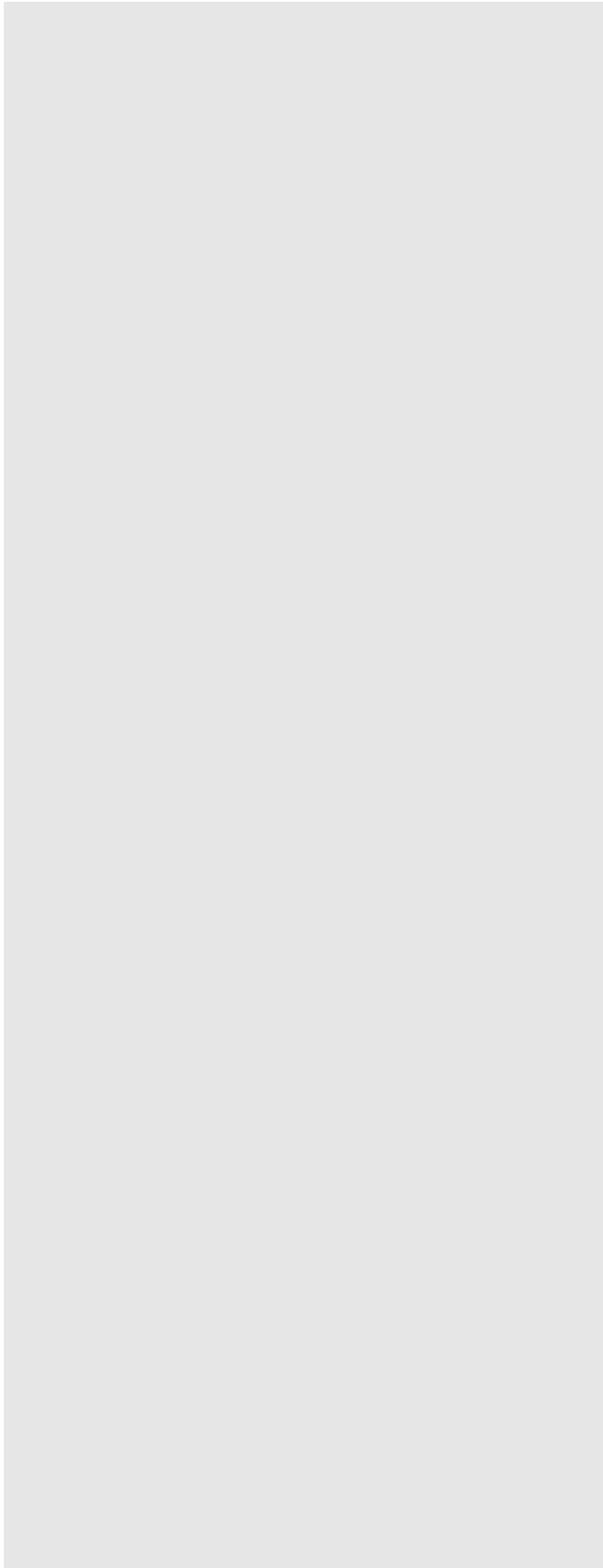
•There must be at least one valid broadcast ephemeris file (EPP file) that spans the entire period. Precise ephemeris files (SP3 files) cannot be used.

•Events in station files (STA files) are not used.

•Geoid undulations can be applied. See *Section 2.5.3.6,* on *Page 28* for details on using geoids in RTKNav.

2.8 Real-Time Graphical Output

### 2.8.1 Display Remote Coordinates and Satellite Info

RTKNav provides a number of text and graphical displays which indicate the state of the program's communication parameters, the coordinates of the baselines and the state of the measurements with respect to each baseline. In addition, there are a variety of waypoint calculations and plots available for real-time high precision navigation to pre-determined points and boundaries.
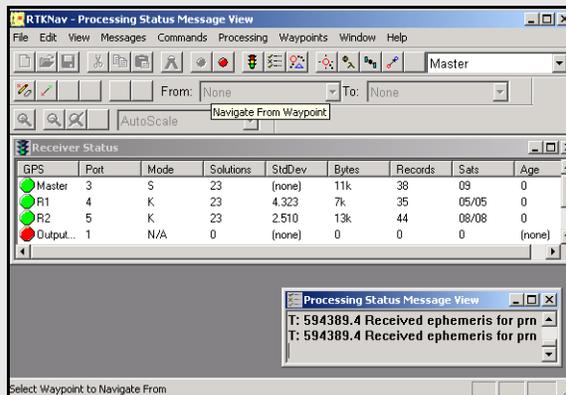

✉   In the toolbar image below, the user has selected *R1* in the list box and used the *Receiver Solution* tool button. The resulting text box is in the screen shot below.
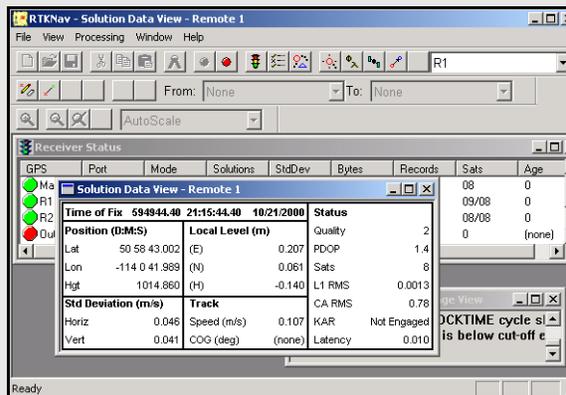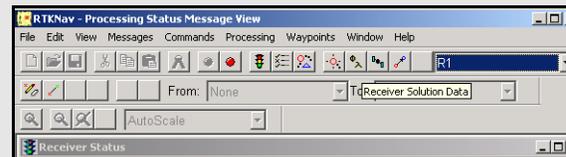
✉   Coordinates of the above include latitude, longitude and height of the remote antenna, as well as delta east, north and up with respect to the base station. In Moving Baseline mode, only the Local Level coordinates are meaningful and of high accuracy due to the nature of relative positioning.

Locktime, azimuth and elevation information can also be displayed (shown on next page)
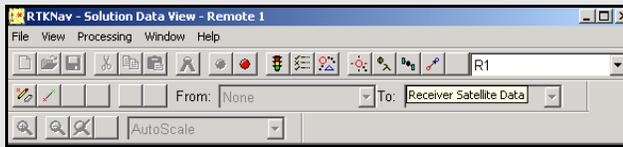
⊠ Again, note that R1 has been selected in the list box and the *Receiver Satellite* tool button. These choices add the text display in the screen shot below.
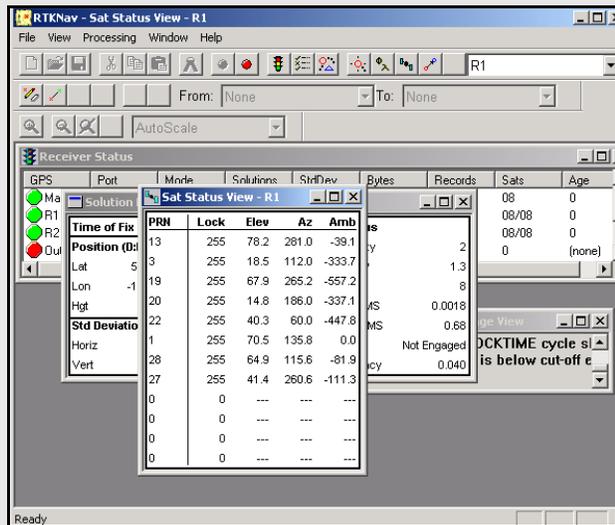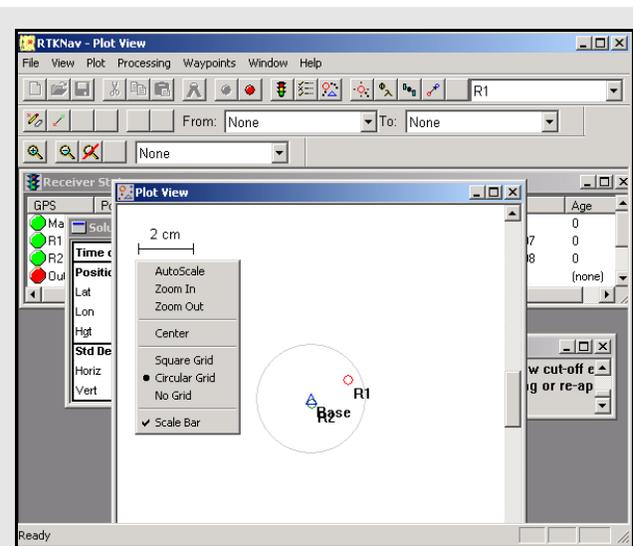

⊠ In the screen box seen above, the azimuth and elevation of the satellites are shown. Of importance in kinematic surveys are the lock values**.** Here, the lock number goes to 255 and stays at this value until phase lock is lost. If lock is lost on any satellite, the lock value goes back to zero and begins to count up with time. In dynamic surveys, the lock values are an important quantity to observe as accuracy depends on continuous phase lock being maintained.

2.8.2 Position Plots of Base and Remotes

Activate the *Plot View* window via *View | Plot* or click on the *Position Plot* tool button. An example of this plot is shown in the screen shot at the top of the shaded box.

To bring up the floating menu, right-click on the *Plot View* window. Right-click on the symbols in the plot window to bring up a menu that allows you to edit the properties of the symbols as seen in the image below. An example of this is in the second screen shot in the shaded box.

2.9 Waypoint Navigation

2.9.1 Defining Waypoints and Boundary Plots

RTKNav allows you to load a list of waypoints and/or enter and edit waypoints interactively. These are pre-determined points that you may wish to navigate to or alternately avoid in real-time. Another use in deformation monitoring is to visually inspect the movement of an antenna in real-time with respect to the a priori coordinates of an object.

Boundaries can be defined by choosing certain waypoints as vertices of a polygon. Waypoints can be defined in the following ways:

- Loaded from a simple ASCII text file
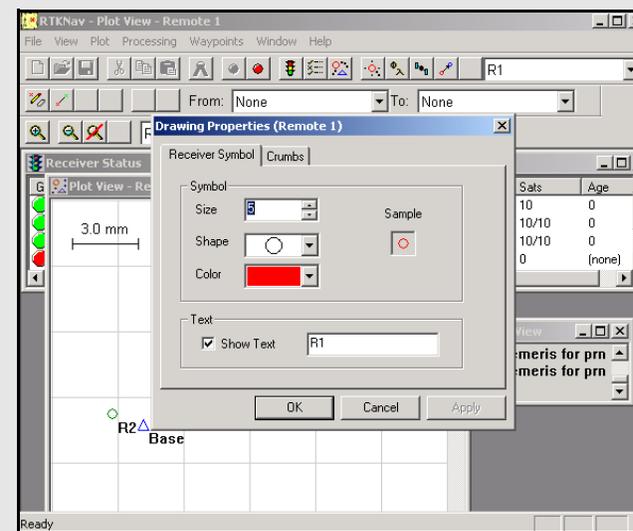- Edited interactively
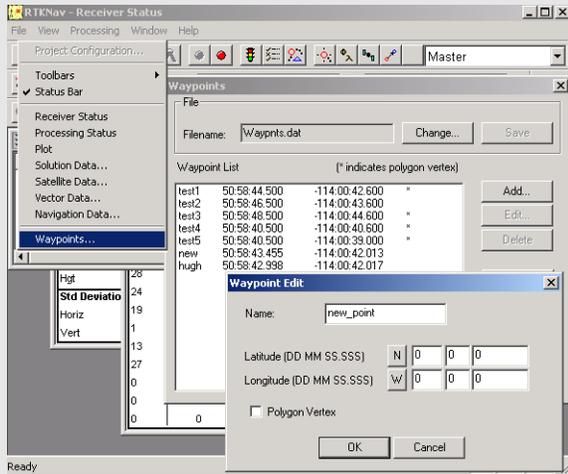- Marked in real-time

2.9.2 Loading Waypoints from the Menu

The screen shot in the shaded box shows what occurs on selecting the *View | Waypoints…*. In this case, a simple waypoint file called WAYPNT.DAT has been loaded from the hard disk.

Waypoint files have the formats listed in the shaded box on *Page 34*.

2.9.3 Displaying Waypoints

To display the waypoints that you have loaded or added, select *View | Plot* or click on the *Plot Position* tool button.

✉ The above is really a zero baseline, with a scale of 2 cm. The base, R1 and R2 are really computing data from the same antenna. The variation in coordinates is just receiver noise.

Waypoint File Formats

(WAYPNT.DAT is used as an example)

- Name latitude (degrees minutes seconds) longitude (degrees minutes seconds) *(optional)

- test1 50 58 44.500000000009-114 0 42.599999999982,*

- test2 50 58 46.5000000000016-114 0 43.599999999998

- test3 50 58 48.499999999997 -114 0 44.600000000065,*

- test 4 50 58 40.499999999996 -114 0 40.600000000001, *

- new 50 58 43.455012698966 -114 0 42.012714166015

- hugh 50 58 42.998050950986 -114 0 42.017246776385

✉ The "*" designates the vertex of some point defining a boundary. RTKNav joins these points with lines to indicate boundary limits, provided the waypoints are actually points on the boundary line that you are interested in. An example might be the limits of an airplane test range. This boundary could be used for range safety purposes.

After displaying your waypoint list in the *Plot View* window, you can also join any two lines in the above plot or you can define boundary vertices and join any polygon defining the boundary. Be sure to enter your polygon vertices in order to define the boundary plot correctly**.** To define a boundary plot make sure that a "*" follows the longitude entry in the waypoint file. See the note in the shaded box.

2.9.4 Displaying Boundary Lines in the Waypoint Plot Window

Displaying a boundary consists of joining the points defined as polygon vertices in the waypoint list. This can be done via *Waypoints | Connect Polygon Points*.

Similarly, any two points can be joined via *Waypoints | Draw Line Between Selected***.**

2.9.5 Marking the Current Remote Point as a Waypoint

If you are interested in marking points as you go, mark the current remote receiver position as a new waypoint. This is useful if you want to save the coordinates of any particular point. To do this, use the *Waypoints | Mark…* option.

3.1 Utilities Overview

This chapter describes the following utilities that are included with RTKNav:

- GPB Viewer

- Concatenate, Splice and Resample

- GPB to RINEX Converter

- GPS Data Logger

This chapter goes through each menu of their interfaces. Step-by-step instructions for first time users are also included.

3.2 GPB Viewer Overview

GPB files are in a binary format and cannot be viewed with a normal text editor. *GPBView*er allows you to both view and edit your data in the GPB View window.

3.2.1 File

3.2.1.1 Open

Any GPB file can be opened with this feature.

✉ If the GPB file is corrupt, *GPBViewer* prompts for permission to fix the file.


3.2.1.2 Close

This feature will close the GPB file without exiting from *GPBViewer*.

3.2.1.3 Saving a GPB File

To make permanent modifications to a GPB file, save a copy the original file. *File | Save As…* gives saves a copy of the current GPB file, and its associated EPP file, under a new name. This lets you edit the new file without losing the original file.

Use this feature to save a file from a newer format into an older format, making it compatible with older versions of the software.
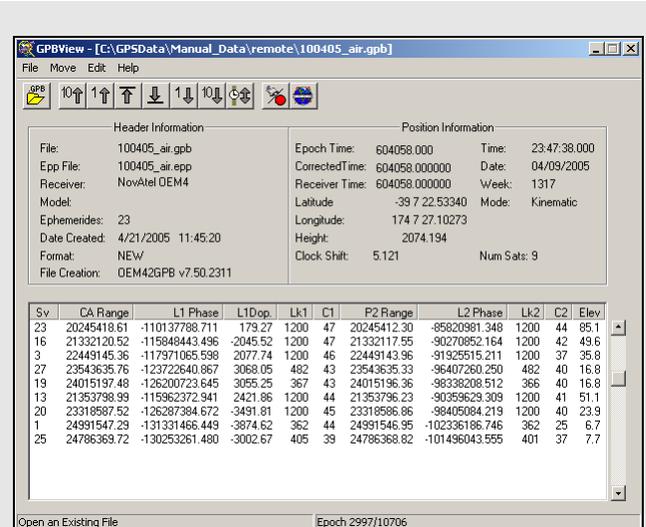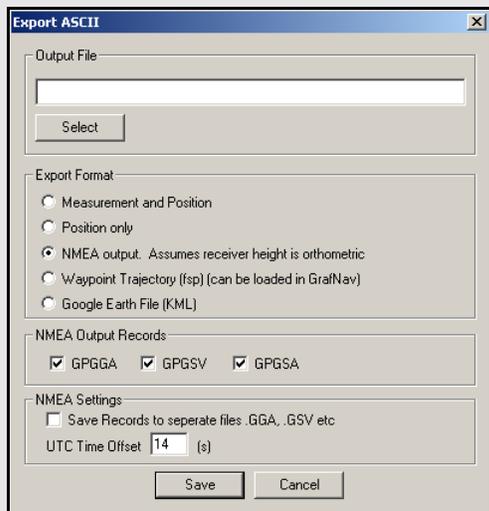
ASCII formats for saving files

Measurement and Position
Exports time, date, raw measurement data and position information for each epoch.

Position only
Exports time, date and position information for each epoch.

NMEA output
Exports data from every epoch in the well-documented NMEA format.

Waypoint Trajectory
Exports the data in Waypoint's single point trajectory (FSP) format. This trajectory file can then be loaded into GrafNav as a solution. See *Waypoint File Formats* on *Page 34* for a definition of the format.

Google Earth
Exports position information into a KML file, which can then be plotted into Google Earth. If you are hoping to download nearby base station data, this feature is very helpful when used in conjunction with the KML files produced for each available download service.

3.2.1.4   Export ASCII

This feature saves data from the binary GPB file into various ASCII formats.

Output File
This option allows you to name the output file. Provide a filename extension because the software will not automatically assign one.

Export Format
The five choices of ASCII formats to save your file in are listed in the shaded box.

NMEA Output Records
This is available for exporting into the NMEA format and allows you to choose which NMEA strings to write to the output file.

NMEA Settings
This is available for exporting into the NMEA format. Enabling the *Save Records to separate files* option writes individual files for each NMEA output record selected under *NMEA Output Records*. You can also enter the appropriate *UTC Time Offset* with respect to GPS time.

3.2.1.5   Load Alternate Ephemeris File

If you lack ephemerides to cover the entire observation period, *GPBViewer* displays a warning message when opening the GPB file. Without a valid EPP file, many features are disabled, including the display of elevation angles and the re-computing of positions. This feature allows you to specify an alternate EPP file for use in the *GPB Viewer*. You can see how many ephemerides have been loaded with the *Ephemerides* field of the *Header Information* box in the top half of the *GPBView* window.

3.2.1.6   Exit

Exits the program.

3.2.2   Move

Forward n and Backward n
Scrolls through *n* epochs in the direction indicated.

Start of file and End of file
Moves to the first and last epoch in the file.

⊠ It is easier to scroll through the GPB file using the shortcut keys, specified under the *Move* menu beside each option.

Search
Moves to a specific location in the file. You can specify an epoch number or a time, in either GPS seconds of the week or GMT format.

3.2.3   Edit

Several options under this menu make permanent changes to the GPB file. You are warned when one of these options has been selected so that you can save a copy of the original GPB file before using it.

3.2.3.1   Switch Static/Kinematic...

If you are unable to switch your data between kinematic and static mode during logging, do it post-conversion with the *GPB Viewer*. This is important for processing kinematic data.

⊠ The status is indicated in the top half of the *GPBView* window, under the *Mode* field of the *Position Information* box.
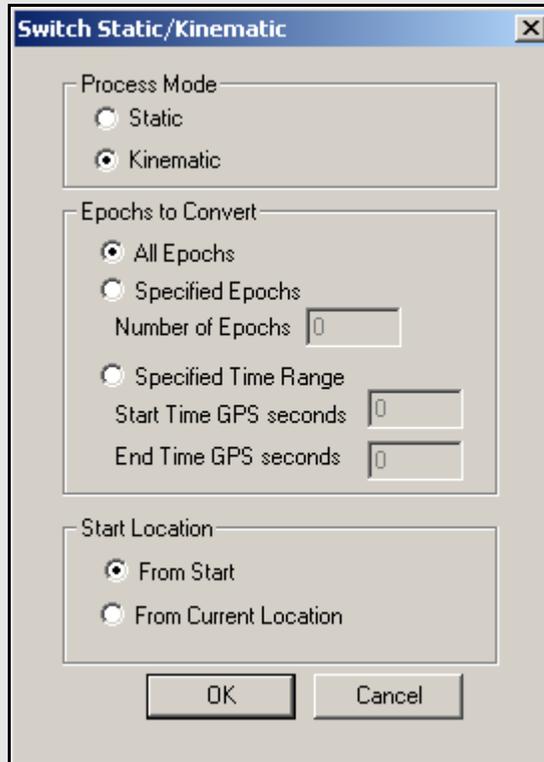

Process Mode
Specifies whether the mode is to be set to *Static* or *Kinematic*.

Epochs to Convert
Determines which observations will be switched. Convert using the options listed in the shaded box.

Start Location
Use in conjunction with *All epochs* and *Specified Epochs*, under *Epochs to Convert*. Determines the starting point of the conversion.


Epochs to Convert options
All Epochs
Switches all epochs from the start of the file onwards or from the current location onwards, depending which starting point is specified under *Start Location* options.

Specified Epochs
Converts the specified number of epochs, subject to the chosen starting point. You can also convert a specific time range that is based on GPS seconds of the week.
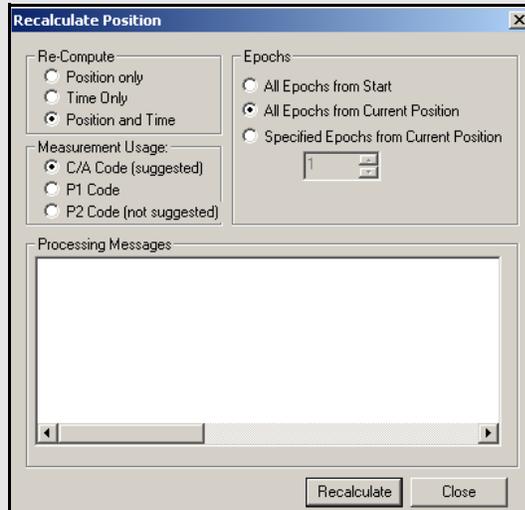
When to recalculate position and time

- To compute position records when they are not being logged because both GrafNet and GrafMov require the GPB file to contain positions.

- To correct erroneous clock shifts.

- To specify a time range to recalculate specific data. You are given the choice as to whether to recalculate only the positions or clock shifts, or both.

3.2.3.2 Recalculate Position and Time

Useful applications for this feature are listed in the shaded box. It removes or induces a cycle slip on any or all satellites at a given epoch. If you add a cycle slip, the locktime for the specified satellite(s) is set to zero. Do this at any epoch where a locktime is greater than zero.

3.2.3.3 Add/ Remove Cycle Slips...

Only remove cycle slips at epochs with a zero locktime and a valid locktime for the satellite of interest at the previous epoch. This feature increments the locktime for the current epoch until the next loss of lock.

> ⊠   Use this feature carefully because the changes can cause problems during processing.

3.2.3.4 Disable Satellite(s)

If a data record is corrupt, eliminated it with this feature. When this option is engaged, the program sets the pseudorange to zero and removes the satellite from any processing that takes place using this file.

3.2.3.5 Recalculate Doppler Measurements

Use this feature if the velocity output from processing appears to be erroneous, indicating possible corruption in the Doppler measurements. The methods available for this recalculation are dependent on the mode in which the data was collected.

3.2.3.6 Align Epoch Time

This feature specifies the data interval and is for data sets where the epoch times are not falling on the proper interval. When this option is activated, *GPBViewer* shifts the epoch time to the nearest epoch, based on the specified interval. Use this feature if the software cannot find any overlapping data between rover and base station files that were logged concurrently.

3.2.3.7 Edit Meteorological Values

This feature edits the GPB file pertaining to the environmental conditions for better troposphere modeling. A TRP file is created, and the processor looks for it when processing. Information about the new troposphere parameters appear in the message logs (FML/ RML) after processing. Enter meteorological values for both the master and remote stations. If you do not, the post-processor utilizes the values from one of the stations for both and only the effect of height difference will be accounted for.

3.2.3.8 Edit Satellite Tracking L2C

In order to properly handle carrier phase measurements tracked using L2C (instead of L2P/Y), a correction value must be applied to make the satellite compatible with L2P/Y. In order to apply this correction, it must be known which satellites are tracking L2C. For some formats such as OEMV, Leica 1200 and RINEX 2.11 can insert these flags automatically. However, L2C decoding for Trimble and possibly some other formats is not supported, and this feature permits the user to set the L2C satellites to be marked. It assumes that there are no changes between L2C and L2P tracking during the data. The software will automatically fill the list of PRNs to change by selecting those that are block IIR-M and later.

3.2.3.9 Edit L2C Phase Correction

In addition to proper satellite registration, L2C processing also needs a correction value. The magnitude of the correction tends to be -0.25, 0.50, 0.25 or 0.00 cycles, and it can be stored in the GPB header. The feature permits the changing of the correction or inserting a correction for those files which do not have one already defined.

Some manufacturers may choose to align L2C with L2P resulting in a correction of 0.00. For the NovAtel OEMV, firmware versions 3.0 and 3.1 use 0.5 cycles while future versions (3.2 and greater) will either use -0.25 or 0.00. For Trimble, early versions used 0.5, while later versions will require -0.25.

3.3   Concatenate, Splice and Resample Overview
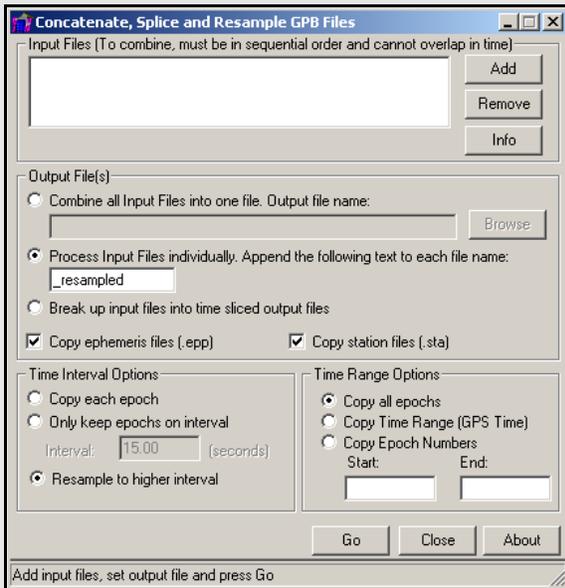
✉ Currently, GLONASS cannot be resampled to higher intervals.

This utility is available in all of Waypoint's programs. This utility joins multiple files together, cuts out a small portion of a file or resamples a file.

3.3.1   Concatenate, Splice and Resample GPB Files

Input Files
Use the *Add* button to locate the GPB files that are to be used. To concatenate several files, add them all at once. The program will sort them chronologically.

Output File(s)
Determines how the creation and naming of new files is handled. For concatenating files, use the *Combine all Input Files into one file* option and provide a name for the output GPB file. For resampling or splicing multiple files, use the *Process Input Files individually* option. The name of the created output files depend on the name of their respective input file and the suffix that is specified. To break up a file into multiple files of *n* minutes, enable the *Break up input files into time sliced output files* option.

Enabling the *Copy ephemeris files* and *Copy station files* options creates output EPP and STA files identical to those associated with the input GPB file.

Time Interval Options
The three options are the following:

> *Copy each epoch*
> Select this if the data rate of the output file is to match that of the input file.
>
> *Only keep epochs on interval*
> Use this when a file is resampled to a lower data rate. The interval specified determines which epochs are copied into the output file.
>
> *Resample to higher interval*
> Use this when a file is resampled to a higher data rate.

✉   Resampling should only be performed on static data. Vehicle motion is not well characterized by the polynomials used so its interpolation is unreliable.

Time Range Options
Determines the range of time that is to be used for the creation of the new file. *Copy all epochs* is generally for resampling purposes. Splicing a file requires the selection of either *Copy Time Range* or *Copy Epoch Numbers*.

Resampling a file prompts the *Resample to a Higher Interval* window, shown the shaded box.

> ✉ Resampling should only ever be performed on static data.

Data Mode
Defines the measurement quantities to be interpolated.

General Options
The two choices for determining the interval of the output file are the following:

> *Interpolate using a Data Interval*
> Defines a specific interval for data resampling.

> *Interpolate using times from a GPB File*
> To have the output file contain data at the same epochs as a different GPB file, select this option. This option is useful for receivers that do not make their measurements on the whole epoch, such as Sirf or Parthus GSU-II.

> ✉ Any gaps in the chosen file is also reflected in the output file.

Advanced Options
The settings are listed in the shaded box.

Ephemeris
Ephemeris data is required to resample a GPB file. Normally, use the EPP file associated with the input. If this file does not exist, or is incomplete, specify an alternate EPP file.

> ✉ Interpolating 30-second data adds 1 to 2 cm of noise. Interpolation of 5-second data or higher adds almost no error. When processing data interpolated from its original interval of 30 second, ARTK may not work. Regardless, when processing, enable the *Only search on exact interval* option and set the *Search on data interval* setting to the data interval of the original source file, under the *ARTK* tab of the processing options.

Advanced options

Window size
Defines the range of data used for interpolation. Should normally be set between 4 and 6 because very large windows could cause problems, especially data files with large intervals.

Number of interpolator terms
Defines the number of Chebychev polynomial coefficients used for the interpolation. Normally, a value close, but not larger than, the *Window Size* is acceptable.

Maximum time span
Prevents the interpolator from using data across too high of an interval. Currently, three minutes is the default but higher values are required for large interpolation windows with coarse input data rates.
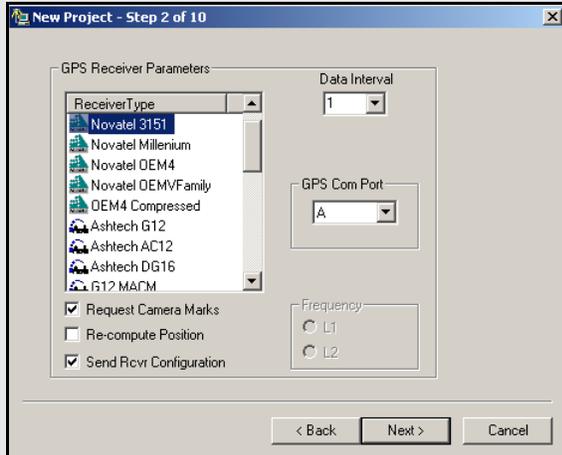
Fine Cycle Tolerance
If a cycle slip larger than the value specified here is detected, then the interpolation is not performed on this satellite.

Coarse Cycle Tolerance
Cycle slips can cause the interpolation to have problems so a coarse detector is implemented. This value should not be lowered too much because pseudorange noise could be mixed up with carrier phase errors.

Code Rejection Tolerance
Any satellite with a code residual greater than the value specified here will not be interpolated. This value may need to be increased for noisy data sets, or for receivers which do not make accurate pseudorange measurements.
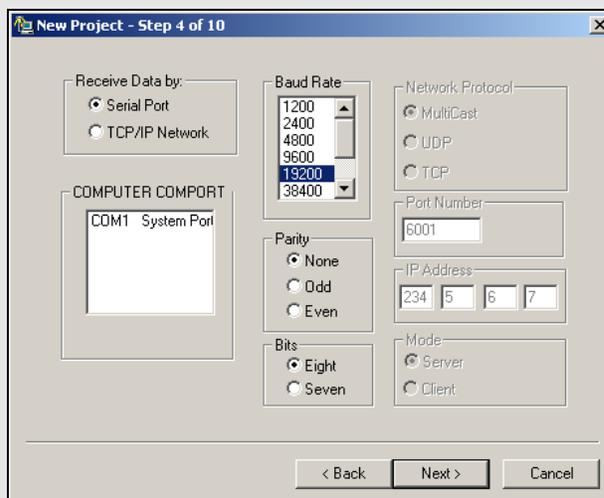
3.4 GPS Data Logger Overview

All of Waypoint's software packages include *WLOG*, a WIN2000/XP/Vista data logger that supports various receivers from NovAtel, Ashtech/Thales, Trimble, CMC, Javad/Topcon, CSI, Navcom, and Ublox. See *Table 1* on *Page 13* for more information on supported receivers.

WLOG configures the GPS receivers, logs the measurement data, and converts it into Waypoint's proprietary format. Adjunct features include waypoint navigation, as well as satellite and scatter plotting capabilities. In addition, stations and events can be marked and written to STA files, which are compatible for post-processing in Waypoint's software.

3.4.1 Getting Started with WLOG

This section examines all the steps required to begin logging data through *WLOG* by analyzing each screen encountered when creating a new project.
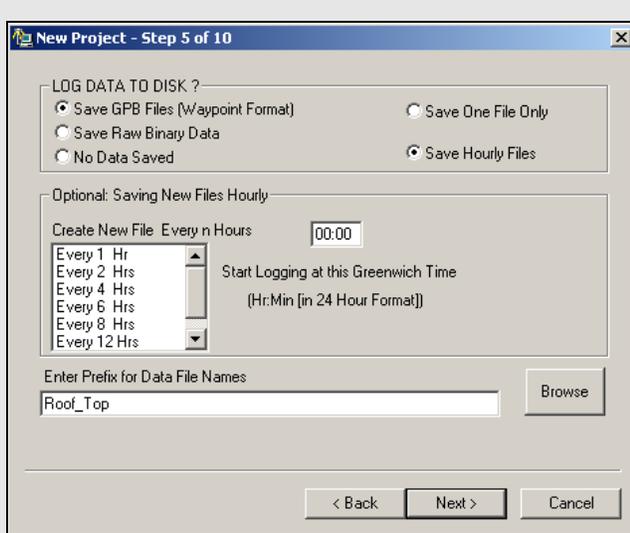
How to log data with WLOG

1. Select *File | New Project* and give the project a name. Click *Next*.

2. Select the appropriate receiver type.

3. Select an interval at which to record the data.

4. Specify the COM port that is being used on the receiver to send the data.

5. For camera marks, enable the *Request Camera Marks* option.

⊠ This option is only available for use with NovAtel, Ashtech and Javad receivers

6. Enable the *Re-compute Position* option to force *WLOG* to output positions at the interval specified in *Step 3*. This alleviates the issue concerning receivers that only output position records on specific intervals.

7. Click *Next*.

8. If the GPS receiver is sending the data directly to the COM port of the computer, select *Serial Port* under *Receive Data by* and proceed to *Step 9*. If the data is being received over a network, select *TCP/IP Network* and proceed to *Step 12*.

How to log data with WLOG cont.

9. If using serial communication to obtain data from the GPS receiver, specify the computer COM port to which the receiver is connected to.

⊠   Choose an appropriate baud rate from the list. The baud rate must be compatible with that of the receiver, and must take into account the amount of data being received that is, size of data records, data logging rate, and so on. Consult the receiver's manual.

---

10. Select the parity. For most receivers, *None* should be selected. Consult the receiver's manual.

11. Select the bits. For most receivers, *Eight* should be selected. Consult the receiver's manual. Click *Next*. Proceed to **Step 15**.

12. If receiving the data by TCP/IP, specify the network protocol and the port number to be used.

13. If using MultiCast protocol, type in the group IP address.

⊠   UDP and MultiCast protocols are ideal for communication over local networks that are within the same office, while TCP is recommended for wide-area communication. Be sure to select a port number that does not conflict with those reserved for standard network purposes. In general, numbers starting at 1024 are usually safe. For MultiCast groups, both the sender and the receiver must have the same IP address. For Win32, MultiCast IP addresses range from 224.00.00.00 to 240.00.00.00.

---

14. Click *Next*.

How to log data with WLOG cont.

15. Select how the data will be saved.

✉   Data can be saved byte-by-byte into a binary file. The entire session will be saved into one LOG file, which will have to be converted into GPB format before being post-processed.

Most users will find it preferable to log directly into Waypoint's proprietary format, because it saves the trouble of converting the data later on when it is time for post-processing. This option also allows for the flexibility of breaking up the data into separate files of *n* hours.

16. Any data being saved to disk will be given the filename specified under *Enter Prefix for Data File Names*. The file extension is dependent on the selection made in **Step 15**.

✉   Files being created every *n* hours will be given unique suffixes.

17. Click *Next*.
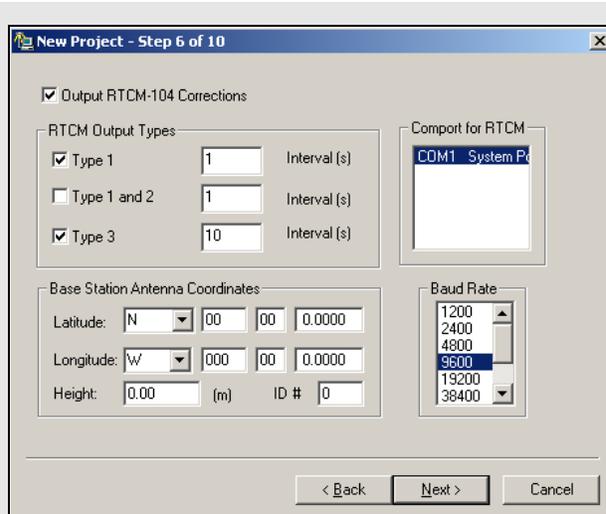
How to log data with WLOG cont.

18. You have the option to select *Output RTCM-104 Corrections*, which are the updated differential corrections. This option must be enabled for the corrections to be displayed.

19. Select the desired type of RTCM output and the interval for each of the following:
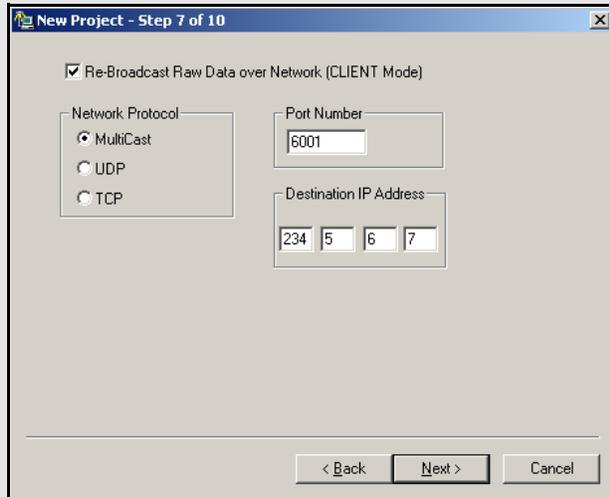
    Type 1: includes time, PRN, $\delta\rho$ and $\delta\rho$ rate.

    Type 2: has the last IODE (issue of data ephemeris).

    Type 3: gives position

    Select the COM port to use for the RTCM output, and enter the coordinates at the base station antenna. Finally, select the baud rate.

20. Click *Next*.
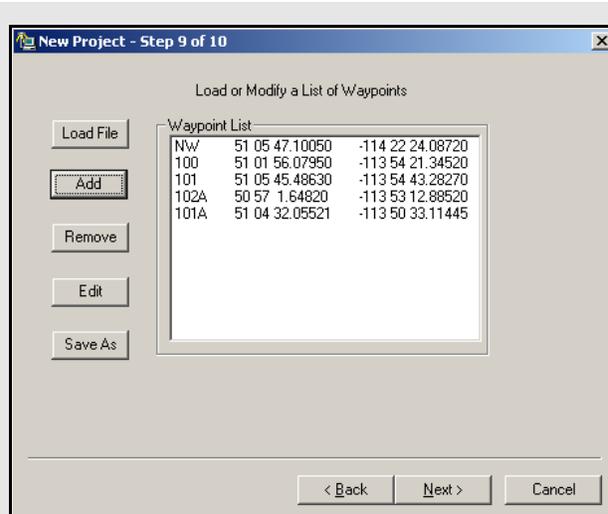
How to log data with WLOG cont.

21. Enable the *RE-Broadcast Data over Network* to re-broadcast GPS data collected by direct link to the GPS receiver over a local or wide-area network.

⊠ This allows two different computers in different locations can collect data from one GPS receiver and view that data in real-time, even though only one computer is directly linked to the receiver. This technique can be used to re-broadcast data using TCP protocol over existing internet connections.

22. Click *Next*.

23. Enter a port number to send out NMEA strings from a secondary serial port of the computer.

⊠ This would be useful in the case where some external device, such as an echo sounder, requires position input from GPS on its serial port.

24. Click *Next*.

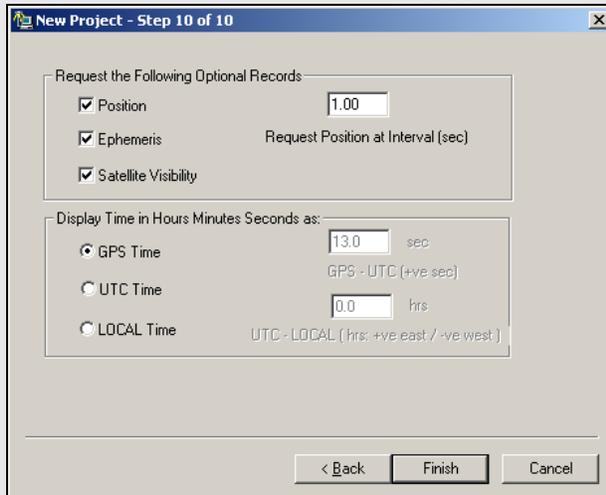## How to log data with WLOG cont.

25. If navigating with a roving receiver, input a list of waypoints for navigation is useful because *WLOG* will provide a text and graphical display of the distance and bearing from the current GPS location to the selected waypoint.

✉   *WLOG* also displays across and along-track information for navigation between any two selected waypoints.

Waypoints can be added in the following two ways:

1. An ASCII file can be loaded, as long as it is formatted correctly. The first element is a station name, followed by latitude and longitude, both in degrees minutes seconds. These fields must be space-delimited.

2. Waypoints can be added the *Add* button. These waypoints can be edited, removed, or saved to a file of their own for later use.

26. Click *Next*.

How to log data with WLOG cont.

27. Make requests for position, ephemeris, and satellite visibility records while logging. You can also display the time in HMS in GPS time, UTC time, or local time. UTC and Local time require the number of leap seconds to be entered as UTC time is a non-linear time system, unlike GPS time.

28. Click *Next*.

If these steps have been followed successfully, *WLOG* will start logging data. If the screen is blank, and the program appears to be inactive, initialization was not successful.

3.4.2 File

This menu allows you to create a new project, open an existing project, load waypoints, find out about the program and exit the program.

3.4.3 Display

This menu gives access to several text displays.

Position
This opens the *Position Information* window, which displays the GPS time, position, velocity, mode, number of epochs that have been logged, the number of ephemerides received, the station name, and the amount of disk space still available.

SatInfo
This opens the *Satellite Information* window, which displays information for each channel that is currently tracking a satellite. The PRN, azimuth, elevation, lock-time, and signal-to-noise ratio is displayed for each satellite being tracked.

Waypoint
This opens the *Waypoint Information* window, which displays information regarding the distance and azimuth between loaded waypoints.

RTCM
This opens the *Differential Corrections* window, which displays information pertaining to the RTCM choices made.

3.4.4 Plot

Satellites
This opens the *Satellite Az/Elev Plot* window, which is a graphical display of the satellites' elevation above the horizon, as well as their azimuths.

Scatter Plot
This opens the *Scatter Plot* window, which is a graphical display of the positions being computed by the receiver. When waypoints have been loaded, this plot is of more use for the remote station, especially waypoints that are represented on this plot by triangles.

3.4.5 Zoom Menu

This menu contains tools to zoom in or zoom out of the *Scatter Plot* window. These tools prove to be particularly useful when a waypoint file has been loaded.

> ⊠ The *Reset Zoom* option will scale the plot back to its default size.

3.4.6 Events Menu

Static/Kinematic *(Shortcut key = F3)*
This feature toggles the mode between static and kinematic. By default, the mode is static so change it for kinematic applications. Failure to do so results in errors during post-processing. Mark any static occupations that may occur during the kinematic survey.

Mark Station *(Shortcut key = F5)*

> ⊠ The toggling of the mode between static and kinematic is considered an event and is therefore automatically recorded in the STA file.

Mark stations that are occupied during the survey with this feature. Stations can be marked while the receiver is static or kinematic mode. Any station marks added are time-tagged according to the GPS time at which the feature was invoked and written to the STA file.

Next Waypoint *(Shortcut key = F6)*
This opens the *Choose Waypoints* window. Choose two waypoints from the list for the distance and azimuth between them and determines the distance from the receiver's current position to a waypoint, and the direction to reach it.
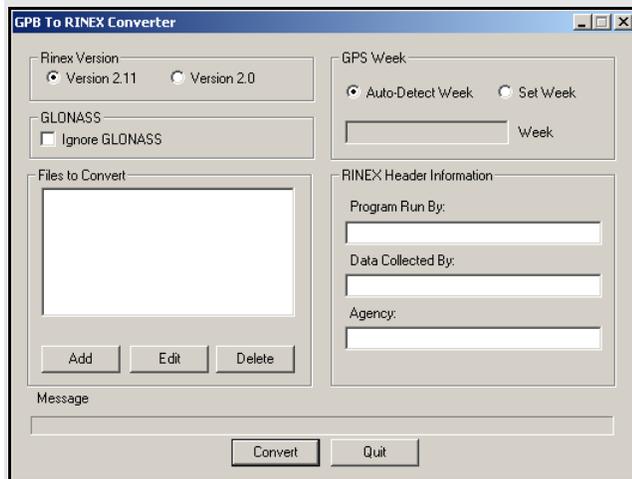
3.5 GPB to RINEX Converter

This utility converts a GPB file into a standard RINEX file, version 2.0 or 2.10.

Files to Convert
Allows you to select the files that are to be

converted.

GLONASS
If present, GLONASS measurement data writes to the observation file, while the ephemerides writes to their own navigation file. This option can be enabled to create GPS-only RINEX files.

Rinex Version
Choose between *Version 2.1* and *Version 2.0* format.

GPS Week
Sets the week number in which the observations were made. If the *Auto-Detect Week* button is selected, then the program determines the week number from the EPP file. Sometimes, an incorrect week number is output to the EPP file during the conversion from a raw data format. In such a case, you must manually enter the week number via the *Set Week* button.

RINEX Header Information
Several fields are available here for editing. This information is used only for the purpose of being written to the header of the RINEX observation file.

Regardless of the option selected, only data measurements that exist in the GPB file are exported so columns of zeroes are not created for missing data.

Output Records

4.1   Output Records Overview

RTKNav can output a number of ASCII NMEA style records. For applications employing more than one remote, the NMEA record(s) are not recommended. This is because they have no identifiers for the remote number. The Waypoint custom records are more suitable since they identify the remote in the record. A summary of the possible records are in *Table 2*.

4.2   Record Format

The format of records follows the NMEA standard. This is an ASCII record protocol using the following definition:

$RecName,v1,v2,v3...*XX

RecName is normally a 5-character record text identifier. Waypoint uses RT to precede its custom records. GP indicates a NMEA GPS record. The items are comma separated. The number of decimals for floating point values is not fixed. Nor is the width of any field fixed. Only the commas should be used to identify separate items. Fields may also be blank (",,") if that value is not available for whatever reason. The end of the data is signaled by an asterix ('*'), which is followed by a two character HEX checksum value. The checksum is a bitwise XOR of each of the characters between (but not including) the '$' and the '*'. Finally, there is a carriage return and line feed at the end.

Table 2: Output Records

Record	Description	When written[a]	Interval[b]
GPGGA	NMEA standard DGPS position record	On Remote	Optional
RTSOL	Geographic position + solution status record	On Remote	Optional
GPAVL	Outputs ECEF position and velocity, as well as local level East, North, and Up among other things.	On Remote	Optional
RTVEC	Base-to-remote local level vector record	On Remote	Optional
RTSLE	RTSOL with some addition status information	On Remote	Optional
RTUTM	UTM remote position record.	On Remote	Optional
RTSIO	Status record of data logging. An interval must be requested for this record	On Interval	Required
RTSAT	Satellite information record	On Remote	Optional
RTBIN	Output of binary solution structure from RtDLL (see rtgnss_baseline_type structure in engine.h). You may need to request this structure from Waypoint. This record is suitable for C/C++ software programmers.	On Solution	Optional
GPVTG	NMEA standard velocity record	On Remote	Optional
RTSTC	Geographic position and fixes	On Remote	Optional
FUGTAR	Fugawi Network Message	On Remote	Optional
RTVECEX	Delta position and velocity and stats	On Remote	Optional

a. This indicates when the results for a record are triggered.
    On Remote = For each remote being processed.
    On Solution = One record for each epoch that is, each time a solution is computed for all remotes.
    On Interval = This record is only triggered on a specific interval.
b. The handling of the data interval field in the LOGREC command or RTKNav dialog boxes.

4.3   Record Descriptions

This section gives a description of each type of record supported by RTKNav. The most commonly used records are RTSOL, RTVEC, RTUTM and GPGGA. See also NMEA-0183 document for the NMEA records that is, GPGGA and GPVTG.

4.3.1   GPGGA Record

This is the NMEA geographic position record (NMEA 0183, Global Positioning System Fix)

Format:

$GPGGA,*hhmmss.ss,lat,N/S,long,E/W,qf,nsats,pdop,ht,M,,,age,# *XX*

Where:

hhmmss.ss Solution time in UTC HMS. May be GPS time if UTC correction not available.

lat	Latitude in (*ddmm.mmmmmm*)
long	Longitude in (*ddmm.mmmmmm*)
qf	GGA quality factor:
	0 NO solution
	0 Single point solution (SPS)
	0 DGPS solution
	0 PPS DGPS solution
	0 RTK with fixed integers (ARTK valid)
	0 Float RTK
nsats	Number of satellites
pdop	Position dilution of precision
ht	Ellipsoidal height in meters (may be pseudo-orthometric if base height entered as ortho-metric).
age	Age of differential solution in seconds.
#	This indicates the remote number (i.e. R1, R2 …).
XX	Checksum (hex).

Example:

```
$GPGGA,171747.00,5058.7165714,N,11400.6999986,W,5,6,1.53,1015.065,M,,,0.1,2
*03
```

4.3.2   RTSOL Record

This record outputs the geographic position, velocity and status information for each remote solution. It is one of the most often used records.

Format:

$RTSOL,*gpstime,*R#,*phi,lamda,ht,sdh,sdv,ve,vn,vh,nsats,*S/K,*qf,dd_dop,status, latency*XX*

Where:

gpstime	Solution time in GPS seconds of the week. This position may be latent by the time estimated in the RTSIO records (*latency*).
R#	This indicates the remote number (i.e. R1, R2 …).
phi	Latitude in decimal degrees.
lamda	Longitude in decimal degrees.
ht	Height in meters.
sdh	Horizontal standard deviation (1 sigma) in meters.

sdv	Vertical standard deviation in meters.
ve,vn,vh	East, north and vertical velocity in m/s. Vehicle track can be computed from the *ve* and *vn* quantities (atan2(ve,vn)).
nsats	Number of satellites used in the solution.
S/K	Mode: static (S), kinematic (K).
qf	Quality factor: best (1), worst (6), no RTK solution (7).
dd_dop	Double difference DOP $\approx$ PDOP^2.
status	No solution (0), single point (1), DGPS (2), RTK (3), RTK(errors) (4)
latency	Estimated latency of the last solution (how far behind). This value does not take into account the latency within the GPS receiver and the transmission time of this record.
XX	Checksum (hex). Same definition as NMEA (bitwise XOR between and not including the $ and *).

Example:

$RTSOL,150210.00,R1,50.97861102,-114.01166575,1015.056,0.082,0.113,-0.06,0.00,-0.12,6,K,3,1.94,3,0.040*4C

4.3.3   GPAVL Record

The format of the GPAVL record is as follows:

R#,utc,lat,lon,ht,veast,vnorth,vup,gpstime,xecef,yecef,zecef,vxecef,vyecef,vzecef,checksum

Where:

R#	Remote number (R1, R2…)
Utc	UTC milliseconds of the day
Lat	latitude – degrees
Lon	longitude – degrees
Ht	Height – meters
Veast	Velocity East
Vnorth	Velocity North
Vup	Velocity Up
Gpstime	seconds of the week
Xecef	X ECEF coordinate
Yecef	Y ECEF coordinate
Zecef	Z ECEF coordinate
Vxecef	X ECEF velocity
Vyecef	Y ECEF velocity
Vzecef	Z ECEF velocity
Checksum	Checksum[cr][lf]

4.3.4   RTVEC Record

The RTVEC record outputs the local level vector between the master and remote. This vector is computed as follows:

X,Y	Rotate ECEF master->remote vector to local level using geographic position of master
Z	Difference between remote and master elevation.

This record can be useful for range and bearing applications.

Format:

$RTVEC,*gpstime*,R#,*east,north,up,sdhz,sdv*,S/K,*nsats,qf,status*XX*

Where:

gpstime	Solution time in GPS seconds of the week. This position may be latent by the time esti-mated in the RTSIO records (*latency*).	
R#	This indicates the remote number (i.e. R1, R2 …).	
east	East coordinate value with respect to master station (meters).	
north	North coordinate value with respect to master station (meters).	
up	Vertical coordinate value with respect to master station (meters).	
sdh	Horizontal standard deviation (1 sigma) in meters.	
sdv	Vertical standard deviation in meters.	
nsats	Number of satellites used in the solution.	
S/K	Mode: static (S), kinematic (K).	
qf	Quality factor: best (1), worst (6), no RTK solution (7)	
status	No solution (0), single point (1), DGPS (2), RTK (3), RTK(errors) (4)	
XX	Checksum (hex).	

Example:

```
$RTVEC,151180.00,R1,0.000,-0.002,0.001,0.019,0.024,K,6,1,3*4A
```

4.3.5   RTSLE Record

The RTSLE record contains everything that the RTSOL record has, and it has additional quality control variables inserted at the end.

Format:

$RTSLE,*gpstime,*R#*,phi,lamda,ht,sdh,sdv,ve,vn,vh,nsats,*S/K*,qf,dd_dop,status,*F/L U*,CaRms,L1Rms*XX*

Where:

gpstime	Solution time in GPS seconds of the week. This position may be latent by the time esti-mated in the RTSIO records (*latency*).
R#	This indicates the remote number (i.e. R1, R2 …).
phi	Latitude in decimal degrees.
lamda	Longitude in decimal degrees.
ht	Height in meters.
sdh	Horizontal standard deviation (1 sigma) in meters.
sdv	Vertical standard deviation in meters.
ve,vn,vh	East, north and vertical velocity in m/s. Vehicle track can be computed from the *ve* and *vn* quantities (atan2(ve,vn)).
nsats	Number of satellites used in the solution.
S/K	Mode: static (S), kinematic (K).
qf	Quality factor: best (1), worst (6), no RTK solution (7)
dd_dop	Double difference DOP $\approx$ PDOP^2.
status	No solution (0), single point (1), DGPS (2), RTK (3), RTK(errors) (4)
F/L/U	Fixed integer ambiguity status: fixed (F), float (L), unknown (U)
L1Rms	L1 carrier RMS (m)
CaRms	C/A code RMS (m)
XX	Checksum (hex). Same definition as NMEA (bitwise XOR between and not including the $ and *).

Example:

$RTSLE,150390.00,R1,51.0,-114.0,1014.9991,0.020,0.026,-0.02,-0.05,-0.03,6,K,1,1.83,3,F,0.77,0.0020*3E

4.3.6   RTUTM Record

This record contains the position in the Universal Transverse Mercator projection. Currently, the zone is picked for each remote. Therefore, position near the zone boundary may result in different zones for different remotes.

Format:

$RTUTM,*gpstime,*R#,*east,north,ht,zone,sdh,sdv,nsats,*S/K,*qf,dd_dop,status,*F/L/U*XX

Where:

gpstime	Solution time in GPS seconds of the week. This position may be latent by the time estimated in the RTSIO records (*latency*).
R#	This indicates the remote number (i.e. R1, R2 …).
east,north	UTM easting and northing (m)
ht	Height in meters.
zone	UTM zone number 1-60
sdh	Horizontal standard deviation (1 sigma) in meters.
sdv	Vertical standard deviation in meters.
S/K	Mode: static (S), kinematic (K).
qf	Quality factor: best (1), worst (6), no RTK solution (7).
dd_dop	Double difference DOP $\approx$ PDOP^2.
F/L/U	Fixed integer ambiguity status: fixed (F), float (L), unknown (U)
XX	Checksum

Example:

$RTUTM,151200.00,R2,709766.8985,5651698.0642,1014.9992,11,0.0189,0.0242,6,K,1,146,3,F*37

4.3.7   RTSIO Record

This record is used to obtain the age of the last solution and latency of the remote solutions. Since solutions for multiple remotes are solved simultaneously, the latency will be the same for all. This record can also be used to determine the GPS time of the last incoming record. The STATUS command is usually better suited for this, but RTSIO can be used as well.

Format:

$RTSIO ,*gpstime,age,latency,num,*M,*time,nsats,*R1,*time,nsats,*R2, *time,nsats…*XX

Where:

gpstime	Solution time in GPS seconds of the week. This position may be latent by the time estimated in the RTSIO records (*latency*). May not be on the whole interval.
age	Seconds since last solution computed. Will increase if base or one of the remotes is not tracking or receiving.
latency	Estimated latency of the last solution (how far behind). This value does not take into account the latency within the GPS receiver and the transmission time of this record.
num	Number of stations to follow.
XX	Checksum (hex).

For each station:

M or *R#*	Master or Remote Number.
time	Time of last incoming record (GPS seconds of the week).
nsats	Number of satellites tracked.

Example:

$RTSIO,149935.9,0.155,0.040,3,M,149935.8,5,R1,149935.8,7,R2,149935.8,5*31

4.3.8   RTSAT Record

This record lists the available satellites for a remote. It indicates the number of satellites used for processing and the elevation, azimuth, locktime and ambiguity for each.

Format:

> $RTSAT,*gpstime*,R#,*nsats*,*prn*1,*elev*1,*az*1,*amb*1,*lock*1,*prn*2,*elev*2,*az*2,*amb*2,*lock*2…*XX*

Where:

gpstime	Solution time in GPS seconds of the week. This position may be latent by the time estimated in the RTSIO records (*latency*).
R#	This indicates the remote number (i.e. R1, R2 …).
nsats	Number of satellites used in the solution (4 values per satellite to follow)
XX	Checksum (hex).

For each satellite:

prn	Satellite PRN number
elev	Satellite elevation in degrees
az	Satellite azimuth in degrees
amb	Satellite ambiguity in cycles (only shows last one thousand cycles –1000.0 to 1000.0)
lock	Seconds since the last cycle slip (may be 255 max).

Example:

> $RTSAT,149580.00,R2,5,25,67,148,0.0,255,29,30,77,350.3,255,21,11,129,706.4,255,11,35,227,474.9,255,30, 21,58,-666.4,255*31

4.3.9   RTBIN Record

This record delivers the binary solution structure for a given epoch. This structure contains the solutions for all of the remotes. Because this structure is quite large (especially for 20 remotes), RTBIN is mainly suitable for TCP/IP communications. The definition of this structure (called rtgnss_baseline_type) is given in engine.h, and comes with the developer's kit. However, you may request access to this structure without purchasing the kit. The structure is packed using a simple algorithm to minimize bandwidth. This record is only suitable for software developers.

> ✉ The rtgnss_baseline_type structure is packed on the one byte boundary (/Zp1).

Format:

> $RTBIN,*NumRemotes*,*MaxRemotes*,*StructSize*,*DataSum*,*Data*…*XX*

Where:

NumRemotes	Current number of remotes in use. Value is in HEX.
MaxRemotes	Maximum number of remotes allowed with this version of RTKNav. Value is in HEX.
StructSize	Number of bytes in unpacked structure. This can be used as a check against your structure size and the unpacked buffer should also be this size. Value is in HEX.
DataSum	This is a 16 bit unsigned value stored in HEX. It is a sum of all of the data bytes in the buffer before packing. It can be used as a check to in addition to the NMEA 8 bit checksum.
Data…	See Unpacking Data below
XX	Checksum (hex).

Example:

```
$RTBIN,2,3,13F0,1069,ð13##03<0030C<007L;0241=<00311<0031D<009+@0A<00317<003
Ð07{001E03<00308<007"1A{002601<003:íþ43}I@5Bò‹%¿_\À#À}44WGS3834<00E@¦TX41¿}
```

```
1D14Ä?X41[001C4L;0241<00608@qÖý!43}I@ÝäÙ%¿_\À#àm07Ð·_@§®1Cõ11`¨?Ÿ]¤ùT|¦?ô®0
1è+ ²?âô§08³È|¿07~ Y¯1D_?0F04È':14_?V nkæ14Ö?þ-
wd¬35Û?×.>i10Oæ?IK&Èât¿;ˆÎi„0C‹?<003`]ý—¿02<006`1Ayø?õ¢ö¹08ù„?ü0A¨ ›`éS?£÷—
òÉ43ð?{003CèÖ<006èÖ<00601[0010306<00319™05„4214V1743
<00Dào@{00381DP45#42²¥˜42?"ôu¿_À<005ào@{0038153723J41æ`#43_‹¸àš?,À<005ào@{0
0380B30b1442432Ce43À,m‰_?_À<005ào@{00381Eý0Aª4124_r42_)õ±24¯ŒÀ<005ào@{00381
4ßî"42l02-
43@^…131CoŠÀ<005ào@[0021E®þFFFFK;0241<00608@¿°'"43}I@Y©u24¿_\À#@Íµõ·_@l‹01ý
06ð¥?0CµN—ûW¤?Ê,09õ08—
°?h1C^Ÿ\7BS¿$¢o«Hè ?Ëô±_±š§?3945#14ËxÕ?ä÷31 M34Ú?†ò›_ú23å?³0D-
1D;¤'?l,Œ10|~_?<003À¶˜t¿02<006`1Ayø?ñÝÐ«l•‹?1C>Þv453C5B?"31˜¤ø—
á?{003CíÖ<006íÖ<00601[0010306<00319™05„4214V1743<00Dào@{00381DP45#42²¥˜42#2
40C35Úáu@<005ào@{0038153723J41æ`#43#36K0F_10†@<005ào@{00380B30b1442432Ce43À
Y0E37ß_}@<005ào@{00381Eþ0Aª4124_r42#ã(NÝÖ„À<005ào@{003814ßî"42l02-
43#11À13_@À<005ào@[007EAøSã%©36104135^°É¨361041*F7
```

Unpacking Data:

To pack the data, the following rules are used:

Character Encountered	What to do
Translated characters (expand to one byte):	
0-9,A-F	Hex nibble (always in pairs). Decode hex byte
#	Byte 00
Repeat characters (expand to many bytes):	
<CCR	Up to 15-character repeat. '<' character followed by 3 hex values. First two values (CC) refer to the hex byte value. R is the number of repetitions (0-15)
{CCRR	Up to 255-character repeat. '{' character followed by 4 hex values. First two values (CC) refer to hex byte value. RR refers to repeat count (16-255)
[CCRRR	Up to 4095 character repeat. '[' character followed by 5 hex values. First two values (CC) refer to hex byte value. RRR refers to repeat count (256-4095)
Finished:	
*	Decoding complete. End of NMEA data identifier.
Otherwise (expand to one byte):	
Ch	Read byte directly. For example, character 'u' would translate to a byte value of 117, etc… Use the ASCII table for your computer+character set.

4.3.10   GPVTG Record

This is the NMEA velocity record (NMEA 0183, Global Positioning System Fix)

Format:

   $GPVTG,*COG,T,MCOG,M,SpeedKnots,N,SpeedKmh,K,CRLF*XX*

Where:

COG	True course over ground, 000 – 359 degrees
MCOG	Magnetic course over ground, 000 – 359 degrees
SpeedKnots	Speed over ground, 00.0 to 999.9 knots
SpeedKmh	00.0 to 1851.8 ko/hr
*XX	Checksum

Example:

$GPVTG,240.81,T,,M,8.12,N,15.03,K*63

4.3.11   RTSTC Record

This record gives geographic information with the time in seconds since 1980 as well as statistics. This is suited for applications. Output occurs after every fixed solution.

Format:

$RTSTC,*1980sec,R#,PassFail,lat,lon,ht,fixed_lat,fixed_lon,fixed_ht,sdhz,sdv,rms,rel*XX*

Where:

1980sec	Seconds since 1980
R#	Remote baseline number
PassFail	Passed or Failed String
lat	Latitude in dms (filtered)
lon	Longitude in dms (filtered)
ht	Height in meters
fixed_lat	Latest fixed latitude (dms, unfiltered)
fixed_lon	Latest fixed longitude (dms, unfiltered)
fixed_ht	Latest fixed height (meters, unfiltered)
sdhz	Standard deviation in east + north (meters)
sdv	Standard deviation height (meters)
rms	Fixed solution RMS in meters
rel	Fixed solution reliability in meters
*XX	checksum (hex)

Example:

$RTSTC,735883290.00,R4,PASSED,34 20 00.31919,-118 01 33.59092,1567.935,34 20 00.31919,-118 01 33.59092,1567.935,0.003,0.005,0.017*0D

4.3.12   FUGTAR Record

This is the Fugawi Network Message – Fugtar for Tracker 3

Format:

$PFUGTAR,*latdegmin,N_S,londegmin,E_W,R#,R#,speed,COG,CCCCCCCC*

Where:

Latdegmin	Latitude in degrees decimal minutes
N_S	North (N) or South (S) hemisphere for latitude
Londegmin	Longitude in degrees decimal minutes
E_W	East (E) or West (W) flag for longitude
R#	Remote baseline number
Speed	speed in m/s
COG	Course-over-ground
CCCCCCCC	default

Example:

$PFUGTAR,3420.0905,N,11823.8120,W,R5,R5,000.0,167.0,CCCCCCCC

4.3.13   RTVECEX Record

This record contains delta position and velocity information. The deltas are always with respect to the base station coordinates.

Format:

> $RTVECEX,*GPSTime,R#,DelEast,DelNorth,DelHt,SDevHoriz,SDevVert,VelEast,VelNorth,VelHt,SDevHorizVel,SDevVertVel,SK,NumSats,qf,status,*XX*

Where:

GPSTime	GPSTime in seconds of the week
R#	Remote baseline number
DelEast	Delta Easting in meters
DelNorth	Delta Northing in meters
DelHt	Delta Height in meters
SDevHoriz	Standard Deviation of horizontal position
SDevVert	Standard Deviation of vertical position
VelEast	East Velocity (m/s)
VelNorth	North Velocity (m/s)
VelHt	Height Velocity (m/s)
SDevHorizVel	Standard Deviation of horizontal velocity (m/s)
SDevVertVel	Standard Deviation of vertical velocity (m/s)
SK	Mode: static (S), kinematic (K).
NumSats	Number of Satellites
qf	Quality factor: best (1), worst (6), no RTK solution (7)
status	No solution (0), single point (1), DGPS (2), RTK (3), RTK(errors) (4)
XX	Checksum (hex)

5.1   Introduction

This chapter gives a description of the various components of MultiEngine, instructions on how to install MultiEngine, as well as installation trouble shooting tips.

5.1.1 What is MultiEngine?

MultiEngine is a Windows 2000/XP/NT/Vista real-time GNSS processing program. It has a simple graphical user interface that is easy to run. MultiEngine also has the ability to support one base station receiver and one or more remote station receiver(s).

MultiEngine has the same basic functionality as RTKNav but with a simplified user interface. In addition, MultiEngine can process asynchronous rover data with long delays of up to several minutes.

The differences between MultiEngine and RTKNav are as follows:

- MultiEngine cannot configure a project. Use RTKNav to create an IN file

- MultiEngine has a minimal user interface. Most of the control should be performed via the command port.

- MultiEngine isolates all remotes from each other. Therefore, a problem on one remote cannot affect the other remotes. RTKNav tries to do this as well, but there are certain data errors that may cause solution errors to be printed on the other remotes.

- MultiEngine is more efficient than RTKNav thereby using much fewer CPU cycles. This allows the 20 baseline limit of RTKNav to be exceeded.

- MultiEngine uses less memory on large numbers of remotes.

- MultiEngine will work on a R3 or R20 hardlock key. Run MultiEngineR3.exe for R3 keys and MultiEngineR20.exe for R20 keys.

- MultiEngine has a filtering feature to reject bad pseudoranges using the user command REJECT_PSR = *ON/OFF Rejection Tolerance.*

5.1.2 Getting MultiEngine Started

In order to get MultiEngine running, the following items are required:

- Computer with MultiEngine installed. Should be a Pentium or faster

- At least two serial ports. Unless TCP/IP data input is utilized, MultiEngine requires one serial port for each GPS receiver it is connected to (either directly or via a radio link). An easy way to get started is to use a USB multi-port serial device.

- For the first try, using a direct (serial) connection is easiest. This way any problems due specifically to the radio link can be isolated later.

- At least two suitable GPS receivers are also required. One is the master and the remaining ones would act as mobiles (i.e. remotes). The receivers should have an open view to the sky in order to work with MultiEngine. Placing them in a window may not provide enough satellites for navigation.

- A serial port is also required if you wish to output serial records to another device. Note that this can also be performed via TCP/IP (TCP mode only).

- The hardware key must be connected to the parallel port.

5.1.3 Installing MultiEngine

Before installing MultiEngine, it is recommended that RTKNav is installed. This will install not just the RTKNav software but also the hardlock driver needed to run MultiEngine.

5.2 Using MultiEngine

The following section describes step by step how to start using MultiEngine.

Running MultiEngine

Before processing, the user must start a new project or open an exiting project. To start a new project, user should use RTKNav. RTKNav initially creates two project files, a IN file and a CFG file. The IN file mainly contains information about communication ports. The processing engine

Starting a New Project:

To start a new project please refer to *Chapter 2* on *Page 17* of this manual for more information.

Run MultiEngineR3.exe for R3 keys and MultiEngineR20.exe for R20 keys. Click *Browse*, and select the IN file or project file created using RTKNav.

To start processing, click *Start Processing*.

To stop processing, user may click *Stop Processing* or *Exit*.

RtGNSSx.dll uses the CFG file to perform GPS processing functions. To get started, the user must first define the GPS receiver types and communication parameters. Please review the following sections very carefully, if you are a first time user.

Steps to begin a *New Project* are listed in the grey box.

Options

Since MultiEngine is a simple version of RTKNav, there is no feature in the program to set the options. Options setting can only be done using RTKNav, please see *Chapter 2* on *Page 17* of this manual.

Features

The following describes some of the interface features in MultiEngine:

- **Browse:** Use to select and open an IN project

- **Start Processing:** Start logging and processing

- **Stop Processing:** Stops the processing

- **Exit:** Close MultiEngine

- **GPS:** Displays the Master (M) or Remote (R#).

- **Port:** The port number.

- **Mode:** Static (S) or Kinematic (K) mode

- **Solutions:** Number of solutions calculated

- **Sdev:** Standard deviation

- **Bytes:** Amount of memory used

- **Records:** Number of records received

- **Sats:** Number of satellites

- **Age/Time:** GPS seconds

- **Messages:** Message log

Prior to processing, the receiver icons are RED to indicate that no data is being logged.

If the baseline solution has been successfully computed, the icons will turn GREEN.

Appendix A

Summary of Commands

The following lists the available commands and a short description of their usage. Any subset of the following may be used for your particular input file.

Command	**Description**
ARTK_QUALITY = *value*	This command determines how strict ARTK is in signaling a fixed integer solution as a pass. It should be left at 1, but if ARTK is having trouble computing a fix at all, try 0. If ARTK is computing incorrect fixes, a higher number such as 2 could be used.
AZ_DETERM = ON/OFF/COMPUTEONLY *Distance DistanceSD*	To use this command the MOVING_BASE command must be set to ON. If this command is set to ON a baseline distance in meters and standard deviation in meters must be set. This helps the solution as the baseline distance is used as a distance constraint. If COMPUTEONLY is used then no distance or distance standard deviation is required.
AZ_ENGAGE = *ON/OFF*	To use this command the MOVING_BASE command must be set to ON and AZ_DETERM command must be ON with a correct baseline distance and standard deviation set. This command will use the distance and distance standard deviation set in the AZ_DETERM command to engage ARTK if the distance between the two antennas becomes too large or too small. Using a small standard deviation might cause the filter to reset continuously, resulting in a possible infinite loop or very slow processing. A typical value for the baseline standard deviation is 2cm.
DATUM = *ProcDatum [Wgs84Conv]*	This command is the datum that processing is to be performed in. See RTKNav for a list of available datums. WGS84Conv is the conversion to be used. Use WGS84Conv=AUTO to allow RTGNSS.DLL to select the best datum conversion. Note that datums are read from manufact.dtm and user.dtm.
DISK_WRITE = ON/OFF	Default is OFF. If DISK_WRITE is ON an output trajectory is written to disk.

DOPPLER_SD = *stdev*

The Doppler standard deviation used by the filter, which is used for determining the velocity. The default value is 1.0m/s.

ELEV_MASK = *angle*

Cut-off elevation for excluding satellites from the computations. This value is expressed in degrees. This elevation mask value may have to be increased to 15 degrees or more in high multi-path environments or during ionospheric storms. Some data tracked under 10 degrees may exhibit noise characteristics. Default value is 10.0.

EXTRAPOL_EPOCHS=*MaxSec*

This command enables base station data extrapolation, commonly used to cut down on radio/network bandwidth. This command should be used with either the EXTRAPOL LineUpMode: in the IN file, or one of the following DLL functions RTProcessEpochBuf, RTProcessEpochMsBuf, or RTProcessBaselineBuf. MaxSec is the maximum number of seconds that extrapolation will be done. A conservative rule of thumb is that extrapolation will add 5mm of position error per second. The maximum extrapolation time allowed is 60 seconds.

INPDATUM=*ON/OFF InputDatum Conversion*

ON will use the processing datum as the input datum. If OFF the base station coordinates are entered in the Input-Datum. Conversion is the datum conversion to use from InputDatum to the processing datum.

LOCKTIME_CUTOFF = *seconds*

L1 Phase data will be ignored for the first n seconds after locking onto the satellite. Change this value to 0 to process all phase data or a large value to ignore the first n seconds of lock. This gives the phase lock loop a chance to stabilize. The default value is 4 seconds.

MASTER_POS = *lat. long. ht [hi]*

These are the coordinates of the master station. The latitude and longitude must be entered in degrees minutes seconds with spaces in between. For those only interested in differential vectors between the stations, a rough pseudorange position is sufficient. This can be found using the VIEWGPB utility or selecting interactive input from the first menu and selecting an averaged position. If the antenna height (metres) is not present, this value will be assumed to be zero. Latitude is positive in the Northern Hemisphere and negative in the Southern Hemisphere. Longitude is positive in the Eastern Hemisphere and negative in the Western Hemisphere (Americas). When using the MOVING_BASE command ON no

MASTER_POS coordinates are required.

MOVING_BASE = *ON/OFF*

Moving Baseline Option Only. If ON, this indicates that the base station is in dynamic mode as well as the remote. If MOVING_BASE=OFF, conventional processing is performed.

NUM_DATA_BUFFERS = *30*

Indicates the number of epochs of data that are to be buffered in memory. When using the export functions RTProcessEpochBuf or RTProcessEpochMsBuf the data buffers are used for aligning data from all baselines.

OMIT_SATS = *sv1 sv2 ....*

This is a list of satellites (PRN numbers) to omit from processing. This may need to be done if a satellite is bad. The default is no omissions.

OUT_PREFIX = *file_prefix*

Normally the output file prefixes will be the same as the input CFG file. This can be used to save multiple output file copies of a project. Care should be taken with this command because the user may loose track of the output files.

OUTLIER_DETECTION = *ON/OFF/POSONLY/VELONLY*

This command enables and disables the satellite outlier detection. By default this command is turned ON and it is recommended to leave it on. Turning this command OFF will disable satellite outlier detection. POSONLY will only look for satellite outliers in the position computation. VELONLY will look for satellite outliers in the velocity computation. The POSONLY command could be used if velocity is of no interest to the user.

PCODE_SD = *stdev*

P-code measurement standard deviation in metres. The default value is 2.25 m. P-code processing and P-code data is required for this option. To change this value the USE_DEFAULTARTK_MEASSTD command must be set to OFF.

PHASE_SD = *stdev*

L1 carrier phase measurement standard deviation in metres. The default value is 0.01 m and it is best left there. To change this value the USE_DEFAULT_MEASSTD command must be set to OFF.

RANGE_SD = *stdev*

CA code measurement standard deviation in metres. For noisy or poor quality pseudorange observations, this

	value can be increased. The default value is 2.25m. To change this value the USE_DEFAULTARTK_MEASSTD command must be set to OFF. When using a low cost receiver it is important to increase the RANGE_SD to value around 7.0m.
SLIP_TOL = *CoarseTol FineTol*	This command allows the cycle slip tolerances to be changed. The default coarse(Doppler) tolerance is 20.0 cycles. The default fine tolerance is 0.5 cycles.
SF_ARTK = *ON/OFF*	This command allows Single Frequency ARTK to be enabled. By default this command is set to OFF and it is highly recommended to leave it OFF.
TROPO_MODE = *ON/OFF/AUTO Distance(km)*	This command allows the ARTK tropospheric state options to be changed. By default this command is set to OFF and it is recommended to leave it OFF. Setting this value to AUTO will have the troposhperic state enable and disable as the baseline distance becomes larger then the distance set in this command. By default the auto distance is 20km.
USE_DEFAULTARTK_MEASSTD = *ON/OFF*	This command needs to be set to OFF to override any of the default measurement standard deviations. If using any of the following commands RANGE_SD, PCODE_SD or PHASE_SD then the USE_DEFAULTARTK_MEASSTD command needs to be set to OFF. By default this command is ON. Note, if using a low cost receiver it is very important to turn this command OFF and set the RANGE_SD to value around 7.0 meters.
VERBOSE = *ON/OFF*	This command enables/disables the verbose messaging mode.

Index