



A NovAtel Precise Positioning Product

Inertial Explorer[®] User Guide

OM-20000106

Rev 9

April 2013



Inertial Explorer User Guide

Publication Number: OM-20000106

Revision Level: 9

Revision Date: April 2013

This manual reflects Inertial Explorer software version 8.50.

Warranty

NovAtel Inc. warrants that its GNSS products are free from defects in materials and workmanship, subject to the conditions set forth on our web site: www.novatel.com/products/warranty/ and for the following time periods:

Software Warranty	One (1) Year
Computer Discs	Ninety (90) Days

Return instructions

To return products, refer to the instructions on the Returning to NovAtel tab of the warranty page: www.novatel.com/products/warranty/.

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, Waypoint, Inertial Explorer, OEM6, OEMV, OEM4, GrafNav/GrafNet, and SPAN are registered trademarks of NovAtel Inc.

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

Table of Contents

Software License	5
Foreword	7
Scope	7
How to use this manual	7
Prerequisites	7
Conventions	7
Customer Service	8
Chapter 1 Inertial Explorer	9
1.1 Overview of Inertial Explorer	9
1.2 Getting Started with Inertial Explorer	9
1.3 File Menu	12
1.3.1 New Project	12
1.3.2 Add Master File(s)	12
1.3.3 Add Remote File	12
1.3.4 Add IMU File	12
1.3.5 Load	13
1.3.6 Convert	13
1.3.7 Removing Processing Files	13
1.4 View Menu	13
1.5 Process Menu	14
1.5.1 Process LC (loosely coupled) and TC (tightly coupled)	14
1.5.2 Combine Solutions	26
1.5.3 Solve Boresighting Angles	27
1.6 Settings Menu	29
1.7 Output Menu	30
1.7.1 Plot Results	30
1.7.2 Export Wizard	30
1.8 Tools Menu	31
1.9 Interactive Windows	31
1.10 Processing Window	31
1.11 Help Menu	31
1.11.1 Help Topics	31
1.11.2 NovAtel Waypoint Products Web Page	31
1.12 About Inertial Explorer	32
Chapter 2 Conversion Utilities	33
2.1 Raw IMU Data Converter	33
2.1.1 Waypoint IMU Data Conversion	33
2.1.2 Creating / Modifying a Conversion Profile	34
2.1.3 Sensor Orientation Settings	35
2.1.4 Decoder Settings	35
Chapter 3 Data and File Formats	37
3.1 Data Formats	37
3.1.1 NovAtel's SPAN Technology	37

3.2 File Formats.....	38
3.2.1 IMR File.....	38
3.2.2 DMR File	40
3.2.3 HMR File	42
3.3 Output Files	43
3.3.1 FIL/RIL/FTL/RTL Files.....	43
3.3.2 FIM/RIM/FTM/RTM Files.....	43
3.3.3 BIF/BIR/BTF/BTR Files	44
Appendix A Summary of Commands	45
Index	49

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. 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;
- (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.

NovAtel retains the right to track Software usage for detection of product usage outside of the license terms.

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, see *Customer Service* on page 8.

Congratulations on purchasing Waypoint® Products Group's Inertial Explorer®.

Inertial Explorer is a Windows-based suite of programs that provide GNSS (Global Navigation Satellite System) and inertial data post-processing. This manual will help you install and navigate your software.

Scope

This manual contains information on the installation and operation of Inertial Explorer. It allows you to effectively navigate and post-process GNSS, IMU (Inertial Measurement Unit) and wheel sensor data. It is beyond the scope of this manual to provide details on service or repair, see the *Customer Service* on page 8 for customer support.

How to use this manual

This manual is based on the menus in the interface of Inertial Explorer. It is intended to be used in conjunction with the most recent revision of the *GrafNav/GrafNet® User Guide* found on the NovAtel web site and the corresponding version of Waypoint's Inertial Explorer software.

Prerequisites

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

Operating System

Windows XP, Vista, 7 or 8.

Hard Drive Space

90 MB of available space on the hard disk.

Processor

A Pentium or Xeon processor is required. Simultaneous forward/reverse processing is possible on dual CPU and Xeon systems. At least 256 MB of RAM is also 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 using the program. This manual has been written with the expectation that you already have a basic familiarity with Windows.

Conventions

This manual covers the full performance capabilities of Inertial Explorer 8.50 data post processing software. The following conventions are used in this manual:

-
- ☒ This is a note box that contains important information before you use a command or log, or to give additional information afterwards.
-

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

Customer Service

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

Call: U.S. & Canada 1-800-NovAtel (1-800-668-2835)
China 0086-21-54452990-8011
Europe 44-1993-848-736
SE Asia and Australia 61-400-883-601

Email: support@novatel.com

Web: www.novatel.com

Write: NovAtel Inc.
Customer Service Department
1120-68 Avenue NE
Calgary AB
Canada, T2E 8S5

1.1 Overview of Inertial Explorer

Waypoint Products Group's Inertial Explorer post-processing software suite integrates rate data from six degrees of freedom IMU sensor arrays with GNSS information processed with an integrated GNSS post-processor (same as GrafNav's). Inertial Explorer utilizes strapdown accelerometer (Δv) and angular rate ($\Delta\theta$) information to produce high rate coordinate and attitude information from a wide variety of IMUs.

Inertial Explorer implements either a loose coupling (LC) of the GNSS and inertial data or tightly coupled (TC) processing that uses GNSS data to limit error during periods where satellite tracking is limited or variable (even if only 2 or 3 satellites are visible). It is important to time-tag the inertial measurements to the GNSS time frame during the data collection process. Proper synchronization is vital. Otherwise, the IMU data will not process. In NovAtel's SPAN system, IMU data is automatically synchronized and Inertial Explorer's GNSS decoder automatically extracts the IMU data.

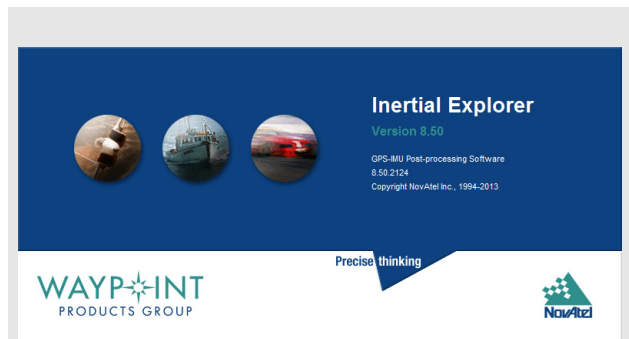
☒ This manual assumes the use of the *GrafNav/GrafNet 8.50 User Guide*. You can request a copy from Customer Service or download it from our Web site at www.novatel.com.

1.2 Getting Started with Inertial Explorer

This section provides a step-by step procedures on how to process data in Inertial Explorer.

Installation

Verify that the installation was successful by ensuring that you have a *Waypoint Inertial Explorer* program group on your computer. If this program group is not there, refer to the *GrafNav/GrafNet 8.50 User Guide* for installation instructions.



How to start Inertial Explorer

1. Verify installation.
2. Click on *Inertial Explorer* to start the program.

How to convert IMU data

1. Open the conversion utility via *File | Convert | Raw IMU Data to Waypoint Generic (IMR)*.
2. Click the *Browse* button to locate the raw IMU data file.
3. Under the *IMU Profiles* box, select the appropriate IMU type.
4. Click *Convert* to create the IMR file. See *Chapter 2, Conversion Utilities* on page 33 for more information.
5. Add the file to the project via *File | Add IMU File*.

How to process IMU Data

1. Click the *Process* menu and then select *Process LC (Loosely Coupled)* or *Process TC (Tightly Coupled)*.

If you are processing in loosely-coupled mode, make sure that you have processed the GNSS data first.

2. Select a processing settings profile based on the IMU type and application.
3. Enter the 3D lever arm offset from the IMU sensor to the GNSS antenna.
4. Add body rotation information if the IMU data was not decoded as Y-Fwd X-Right Z-Up.
5. Click *Process*.

Convert and Process GNSS Data

Refer to the *GrafNav/GrafNet 8.50 User Guide* to process GNSS data. The only exception is that the new project is created in Inertial Explorer, not GrafNav.

For NovAtel SPAN, be sure that the *Extract inertial SPAN data* option is enabled in the OEM4 decoding options so that Waypoint's generic IMR (raw IMU data) file is created automatically.

Convert IMU Data

IMU data must be converted to Waypoint's generic IMR format for processing. To do this, follow the steps in the shaded box.

NovAtel SPAN users don't have to follow these steps because they have a one-step process to convert their IMU data.

Determine Alignment Method

Automated alignment forces the software to scan the data and determine the best alignment method. Only set the alignment options manually if the automated method returns poor results.

Once the GNSS data is processed, use the *File Data Coverage* plot in conjunction with the *Velocity Profile* plot to decide which alignment mode to use. Perform a static coarse alignment for two minutes in both directions for optimal processing unless you are working with a MEMS-based system.

If no static data is present at the start and/or end times of the *Velocity Profile* plot, then perform a kinematic alignment instead. For static alignment, this plot helps to determine how much time is available for the processor to use. The amount of time for the alignment can be set under the *Method for Initial Alignment* options accessed through *Advanced... | System*. The correct time length must be assigned because any movement during the static alignment results in an error.

Fine static alignment is only practical for high accuracy IMUs and generally requires at least 10 minutes. Kinematic alignment requires 4-8 seconds.

Process IMU Data

The steps for processing IMU data are in the shaded box.

Plotting and Quality Control

Once processing is complete, view the quality of the results by analyzing the IMU plots. Under the *Output* menu, choose *Results* to access the following IMU plots:

Attitude (Roll and Pitch)

This plot shows the roll and pitch profile of the processed IMU data.

Attitude (Azimuth/Heading)

This plot shows the heading/azimuth of the IMU and the GNSS course-over-ground. They should be in reasonable agreement.

If the red line and green line are 180° different, then the IMU has been mounted backwards, and a rotation of 180° about the Z-axis will need to be entered under the *Process Tightly Coupled* or *Process Loosely Coupled* settings window.

Attitude Separation

This plot requires that forward and reverse have both been processed. It shows the difference between their attitude values. Ideally, they should agree.

IMU-GNSS Position Misclosure

This plot shows the difference between the GNSS-only and the GNSS/IMU trajectories. These plots should agree.

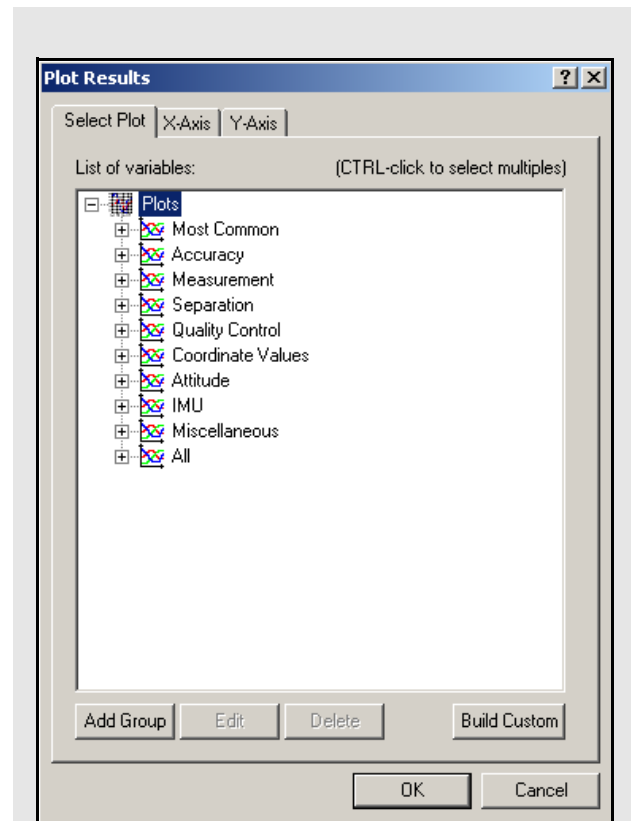
Use the *Build Custom* button to add some of these plots to a customized list.

Smooth Solution

Select *Process | Smooth Solutions* to smooth and combine the two directions. This is strongly recommended in order to achieve the best results.

Export Final Coordinates

The steps for exporting final coordinates are in the shaded box.



How to export final coordinates

1. Select *Output | Export Wizard*.
2. Specify the source for the solution. *Epochs* outputs the trajectory, while *Features/Stations* exports positions only for loaded features, such as camera marks.
3. Select a profile. For Inertial Explorer, select *IMU Data* as the profile.
4. Click *Next*.
5. Use the processing datum for the datum screen.

If prompted for the geoid undulation file, it can be found on the distribution CD or the NovAtel web site.

Project Wizard Steps

1. Create and name the project
2. Add the rover data to the project.

The rover data can be in Waypoint's GPB format or in the receiver's raw format. If the data is in the receiver's raw format, the Wizard converts it to GPB for you.

If you are a NovAtel SPAN user and you add a raw data file, the Wizard automatically detects the IMU model for conversion to IMR format.

3. Add the base station data to project.

You can add your own local base station data (in raw or GPB format) or you can have the Wizard download free service data from the Internet.

If you plan to process with PPP, you can skip the previous step and download the precise satellite clock and orbit files from the Internet.

How to start a new project using *Auto Start*:

1. Enter a name for the project with *File Name*. Check that the file path is pointing to the directory where the project files are saved.

Choosing a name of an existing project overwrites that project.

2. Choose the Master Station File(s). This is the GNSS data file collected at the reference station.

Raw GNSS data files must first be converted to Waypoint's common format (GPB). Refer to the *GrafNav/GrafNet 8.50 User Guide* for more information on converting data.

3. Choose the Remote File. It must contain GNSS data collected during the same time period as the reference station.

Continued in the shaded box on the next page.

1.3 File Menu

Refer to the *GrafNav/GrafNet 8.50 User Guide* for information on the features available via this menu. The points relevant to Inertial Explorer are discussed in this section of the manual.

1.3.1 New Project

Project Wizard

The Project Wizard offers you a guided step-by-step way of creating a project. The *Project Wizard* steps are listed in the shaded box.

Auto Start

Auto Start prompts for all the information required to process IMU data. This option adds reference and remote stations and allows a project setting that best suits the application. The steps for starting a project using *Auto Start* are in the shaded box.

1.3.2 Add Master File(s)

Entering the proper reference station coordinates is essential for obtaining high accuracies. The values that appear are averaged from the GPB file and may have errors of 10 metres or more. For many publicly available stations (i.e. IGS, CORS, etc.), precise coordinates are pre-loaded in the *Favourites Manager*. Alternatively, you can click the *Compute from PPP* button to have the software attempt to solve for the precise coordinates using the PPP engine. Refer to the *GrafNav/GrafNet 8.50 User Guide* for more information.

Proper datum selection is very important as well. WGS84 coordinates are different from NAD83 coordinates by ~2 m, so care should be taken when using these coordinates.

1.3.3 Add Remote File

When adding the remote GNSS data file, leave the antenna height at zero. The vertical offset should be accounted for as part of the IMU to GNSS lever arm.

1.3.4 Add IMU File

When starting a new project, the program needs the data collected from the IMU.

The IMU file must be in the IMR format before being added.

1.3.5 Load

LC Solution (Loosely Coupled)

Loads the loosely coupled solution.

TC Solution (Tightly Coupled)

Loads the tightly coupled solution.

Camera Mount Data

Loads auto-stabilized camera mount information.

1.3.6 Convert

Raw GNSS to GPB

Raw GNSS data must be converted to GPB format for processing. Refer to the *GrafNav/GrafNet 8.50 User Guide*.

Raw IMU Data to Waypoint Generic (IMR)

IMU data must be converted to IMR format in order to be processed by Inertial Explorer. Use this utility to perform this conversion. See *Convert IMU Data* on page 10 for more information.

GPB to RINEX

This converts GPB files to a RINEX file. It supports the creation of Version 2.0 and 2.11 of the RINEX format. For additional information, refer to the *GrafNav/GrafNet 8.50 User Guide*.

1.3.7 Removing Processing Files

This utility removes all the files associated with any given project. Refer to the *GrafNav/GrafNet 8.50 User Guide* for details of this utility. The Inertial Explorer capabilities are discussed here.

Files to Remove

Selects files to remove from the project or folder.

Inertial Processing (LC and TC)

Removes all ASCII and binary files created during IMU processing, including message logs and trajectories.

1.4 View Menu

Refer to the *GrafNav/GrafNet 8.50 User Guide* for a description of all the features available in this menu

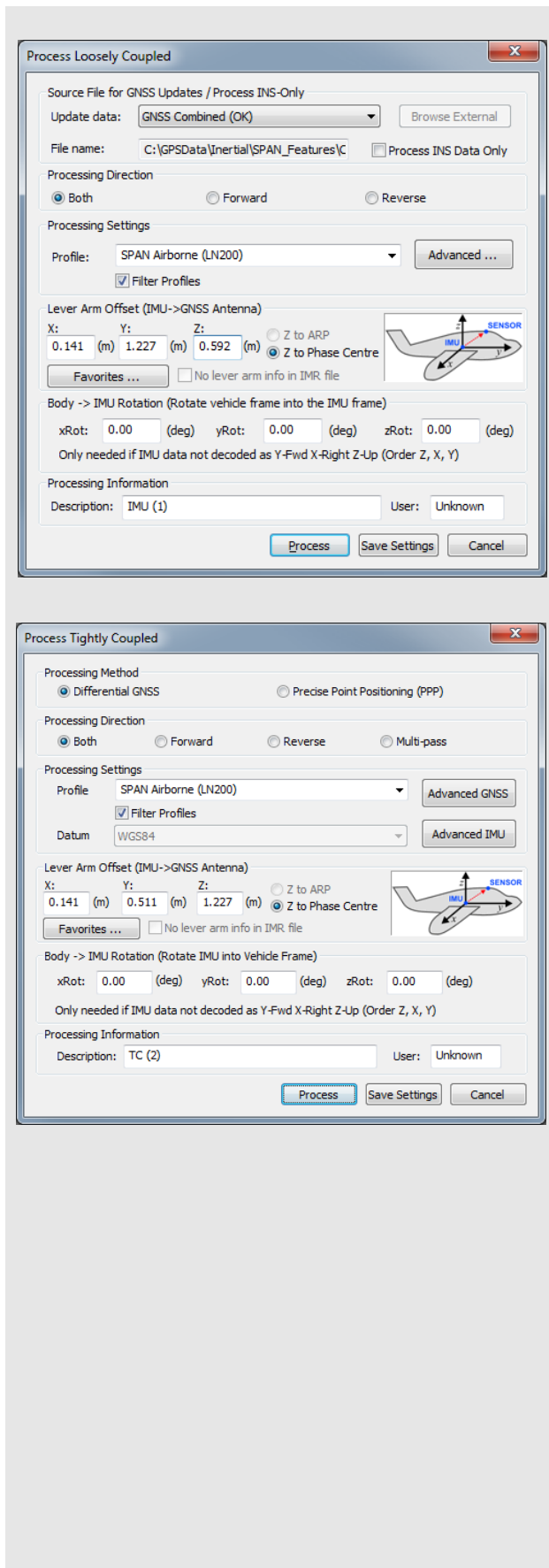
-
- ☒ In Inertial Explorer, view IMU message logs and trajectory files under *View | Forward Solution* and *View | Reverse Solution*. See *Section 3.2, File Formats* on page 38 for information on file formats.
-

How to start a new project using *Auto Start* con't...

4. Choose the IMR file. The IMU data must be concurrent with the master station, base station and remote data. To verify this, click the *Plot Coverage* button to display the *File Data Coverage* plot.

-
- ☒ The IMU data must be converted to the IMR format before being added to the project.
-

5. Choose the DMR file. If any DMI data is available, select the *Enable DMI data* option. This data must be collected concurrently with the rest of the data in the project. Use the *Plot Coverage* button to find overlapping DMR data.
6. Choose a Processing Profile that best suits the application and the IMU being used.
7. Enter the Master Station Coordinates.
8. Enter the Remote Station information.



1.5 Process Menu

Refer to the *GrafNav/GrafNet 8.50 User Guide* for information regarding all of the features available from this menu. Only those features that are exclusive to Inertial Explorer are discussed here.

1.5.1 Process LC (loosely coupled) and TC (tightly coupled)

This window provides access to most settings related to IMU processing.

Source File for GNSS Updates (LC Processing)

Update Data

Use this option to select the GNSS file from which Inertial Explorer obtains updates. In most cases, the combined solution is suggested. However, you may specify an alternate file by selecting External trajectory from the drop-down menu and clicking the *Browse External* button.

File Name

Displays the selected file that will be used for updates.

Process INS Data Only

This option disables the use of GNSS data during INS processing. Updates will only be performed with user-entered coordinate updates.

-
- This mode of processing is not recommended. It is only used for special applications, such as pipeline pigging.
-

Process Settings

Profile

Select an appropriate profile from the drop-down menu prior to processing. This ensures that the GNSS and IMU settings are configured in an optimal way. Use the *Advanced...* button to make adjustments to the settings.

-
- If pre-processing was performed during GNSS data conversion, a profile is automatically selected.
-

Filter Profiles

Enabling this feature ensures that only the profiles specific to your IMU model are shown in the list. This option scans the IMR header to determine the name of the IMU and is largely aimed at NovAtel SPAN customers.

Advanced...(LC Processing)

This button provides access to advanced settings related to INS data processing and lets you choose the options best suited to your application.

Advanced GNSS (TC Processing)

This button provides access to advanced settings related to GNSS processing. Refer to the *GrafNav/GrafNet 8.50 User Guide* for information.

Advanced IMU (TC Processing)

This button provides access to advanced settings related to INS data processing and lets you choose the options best suited to your application.

Lever Arm Offset (IMU → GNSS Antenna)

To perform GNSS updates accurately, enter the 3-D offset, in metres, from IMU sensor array’s navigation centre to the GNSS antenna. This offset vector must be entered with respect to the body-frame of the vehicle, as the image in the shaded box shows.

You must also specify whether the Z value applies to the antenna’s reference point (ARP) or L1 phase center. To specify ARP, you must select an antenna model when you add the remote GPB file to the project. In this case, the antenna model’s offset value is applied to the Z value to raise the Z value to the L1 phase center.

-
- Save lever arms for future access using the *Favorites* button.
-

Read from IMR file

If the lever arm values are written to the header of the IMR file, then use this option to extract them.

Body-to-IMU Rotations (Rotate Vehicle Frame into IMU Frame)

Many typical IMU installations have the surface of the IMU directly attached to the floor of the vehicle so the sensor frame of the IMU and the body frame of the vehicle are more or less aligned. In these installations, the roll, pitch and yaw of the vehicle are directly sensed by the IMU. Some IMUs are installed in a tilted position with respect to the body frame of the vehicle. If the tilt between the IMU frame and body frame is known, Inertial Explorer compensates so that the attitude information produced is with respect to vehicle body frame, not the IMU sensor frame.

The order of rotations employed is R_z , then R_x , followed by R_y , in decimal degree units.

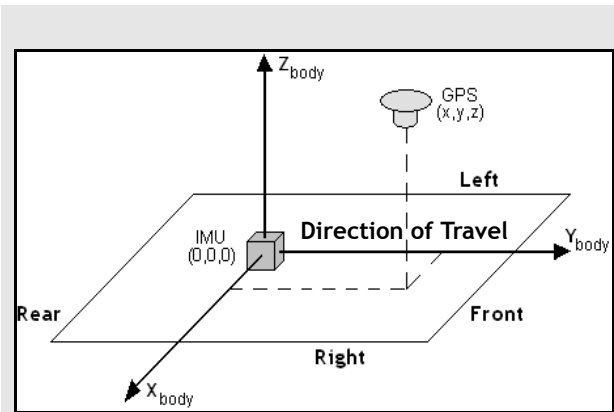
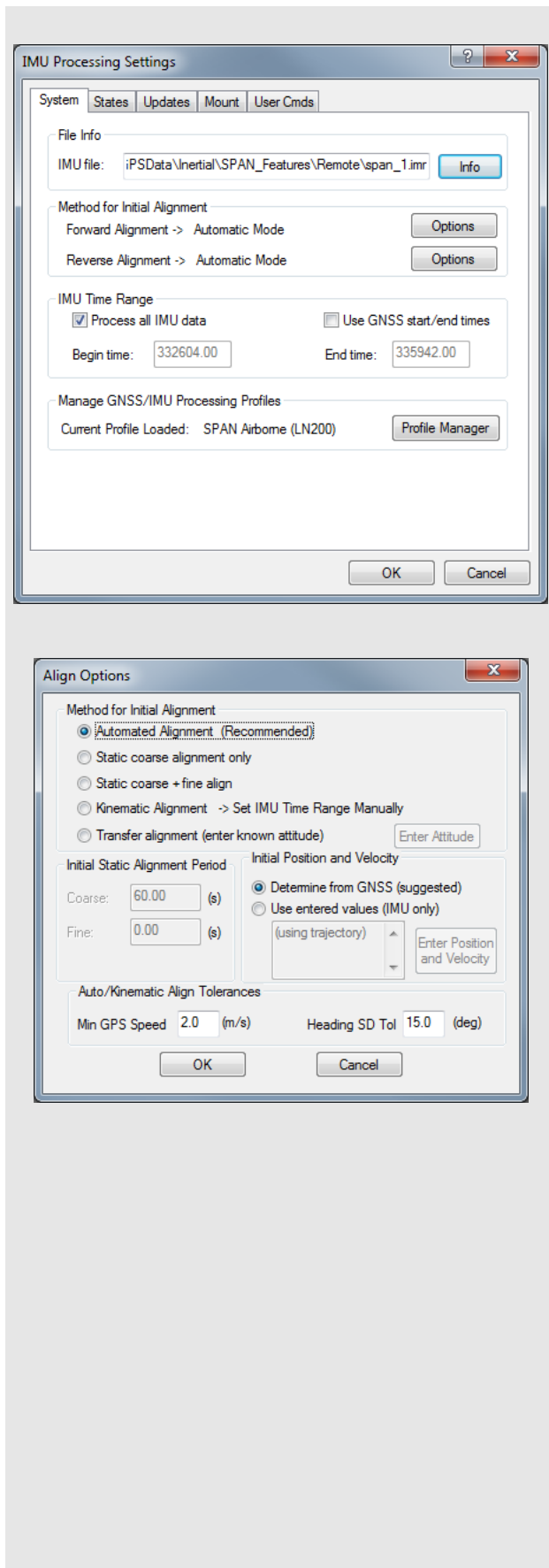


Figure 1: Body Frame Definition for Lever Arm Offset

The IMU is the local origin of the system and the measurements are defined as the following:

- X:** The measured lateral distance in the vehicle body frame from the IMU to the GNSS antenna.
- Y:** The measured distance along the longitudinal axis of the vehicle from the IMU to the GNSS antenna.
- Z:** The measured height change from the IMU to the GNSS antenna.

-
- All measurements are from the navigation center of the IMU to the GNSS antenna phase center.
-



Advanced...

This button provides access to advanced settings related to INS data processing and lets you choose the options best suited to your application.

System

File Info

The following fields are displayed:

IMU file

Displays the path to the binary IMU measurement file that was added to the project. If incorrect, go to *File | Add IMU File* to locate the proper raw data file.

Method for Initial Alignment

Initial alignment, whether processing forward or reverse, determines the initial roll, pitch and yaw of the IMU. Alignment settings should be set with care to ensure that the best possible alignment is formed.

The following options are available and are largely dependent on the length of static data present:

Automated Alignment (Recommended)

This option forces Inertial Explorer to scan the available data and automatically determine what type of alignment to perform. This option is highly recommended. The other options should only be used if automated alignment is failing.

Static coarse alignment only

The data being used for alignment must be completely static as any motion results in an error. Coarse alignment is performed for the first 30 to 120 seconds of static IMU data collection and is used as a seed value for attitude information for either fine alignment or navigation. Coarse alignment uses the sensed gravity vector components to estimate roll and pitch. It uses sensed Earth-rotation rate to provide an initial estimate of the yaw of the IMU.

-
- ☒ Most low accuracy sensors, such as MEMS, can measure gravity components, but the Earth-rotation rate, which is roughly 15 arc seconds per second at the equator, may be masked by gyro noise. As a result, the initial yaw estimate from the coarse alignment may be wrong.

For applications involving constant GNSS updates, coarse alignment is often enough to start kinematic IMU navigation.

Static coarse + fine alignment

Use this option if the IMU is static for longer than 120 seconds.

-
- ☒ Many high precision IMU applications recommend approximately 2 minutes of coarse alignment followed by 8 to 10 minutes of fine alignment.
-

Within 5 to 10 minutes, GNSS updates enable the IMU to provide attitude information consistent with the accuracy level achievable by the accelerometer/gyro triad, with or without fine alignment. This depends on the type of IMU, and the application's requirements. After roll, pitch and yaw are roughly estimated for coarse alignment, fine alignment refines them to a better level of precision.

Kinematic alignment

When neither static data or a priori attitude information are available, alignment can be done using vehicle motion. This requires a short period of time where the vehicle is relatively level and moving in a relatively straight line followed by some higher dynamics, such as an aircraft S turn. The time period can range from 4 to 40 seconds, depending on the dynamics at the chosen start time.

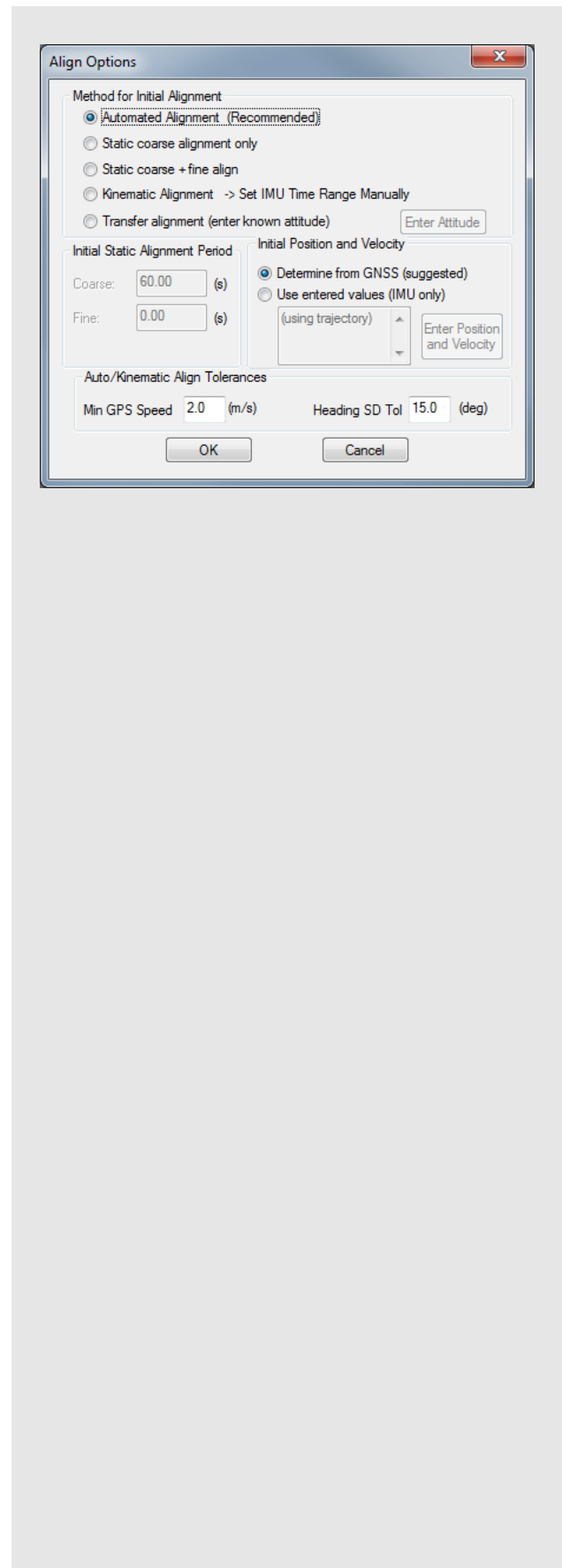
This feature is useful when there is no static data. While Inertial Explorer has been successful at aligning tactical grade systems using this method, there is no guarantee regarding lesser grade IMUs. It might even be necessary to pick a different error model in order to prevent instabilities arising in the Kalman filter.

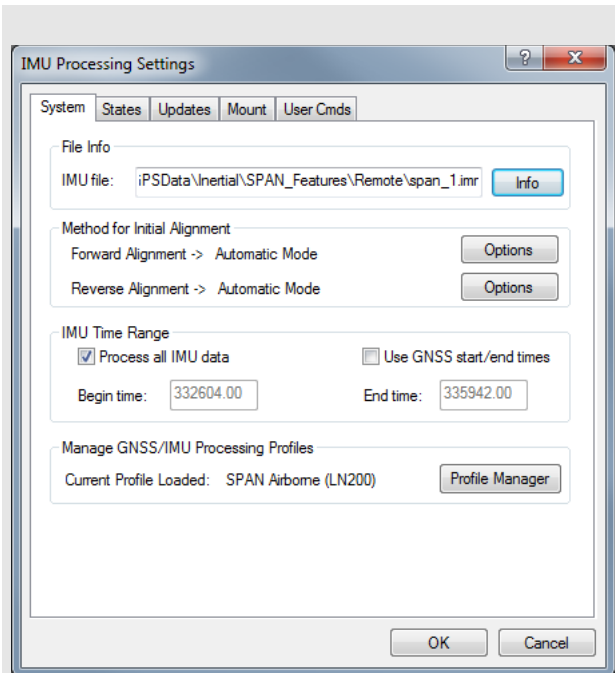
Transfer alignment

If roll, pitch and yaw are known, these values can be entered as initial integration constants to allow navigation to proceed. Attitude angles can be provided by another IMU, in which case the misalignment between the IMUs must be applied, or they can be extracted from another trajectory, such as the opposite processing direction.

Click *Enter Attitude* to enter initial attitude information manually or select *Get from Trajectory* to scan at a specified time from a defined IMU trajectory.

The first epoch appearing in the IMU trajectory file of the direction opposite that of the one being processed is used. The attitude and velocities, as well as their standard deviations, are loaded





IMU Time Range Options

Process All IMU Data

If this option is enabled, the software obtains the beginning and end times from the raw binary IMU file. These times are in GPS seconds of the week for typical GPS time-tagged applications.

Use GNSS start/end times

When selected, IMU processing will start and end based on a time range set under the *General* tab of the GNSS processing options menu.

Begin Time

For forward processing, this time should correspond to a time in the trajectory files created during GNSS processing. If it is, then Inertial Explorer uses the GNSS-derived position at that time to seed the inertial processing.

-
- The time used here is mission-dependent and should be chosen with care because a poor alignment results in poor attitude and coordinate information.
-

End Time

Same considerations as above. This time determines the point at which forward processing stops and when reverse processing begins.

Initial Static Alignment Period

The length of time assigned to static alignment depends on the method of alignment being used. In all cases, it is important that the values entered are in accordance with the *Begin* and *End* times specified under the *General* tab of the GNSS processing options menu.

To perform a static alignment, specify the length of time that the IMU was stationary. If this is unknown, the *Velocity Profile* plot obtained from the GNSS processing is useful.

-
- This field does not apply for transfer alignment or for kinematic alignment.
-

Initial Position and Velocity

The two options include the following:

Determine from GNSS (suggested)

This method is for collected GNSS data in addition to IMU data. The starting position and velocity is read in from the GNSS trajectory specified under the *Source of GNSS Updates* box in LC processing and in the *General* tab for TC processing.

Use entered values (IMU only)

This option is for performing IMU-only processing. If GNSS data has been processed, load the position from a computed trajectory. Otherwise, enter it manually. In either case, click the *Enter Position and Velocity* button to access the input window.

Auto/Kinematic Align Tolerances

Min GPS Speed

This parameter specifies the minimum speed that the system must be traveling before kinematic alignment is attempted. It should only be lowered if the application involves very low dynamics.

Heading SD Tol

This parameter specifies the tolerance below which the heading standard deviation must fall before the alignment routine will move onto navigation mode. Lower this value if the software is not achieving a good alignment. Raise this value if the software is not aligning at all.

IMU Time Range

These options are in the shaded box.

Manage GNSS/IMU Processing Profiles

Current Profile Loaded

Displays the profile that is being used. To see what each profile's configuration looks like, click the *Profile Manager* button.

States

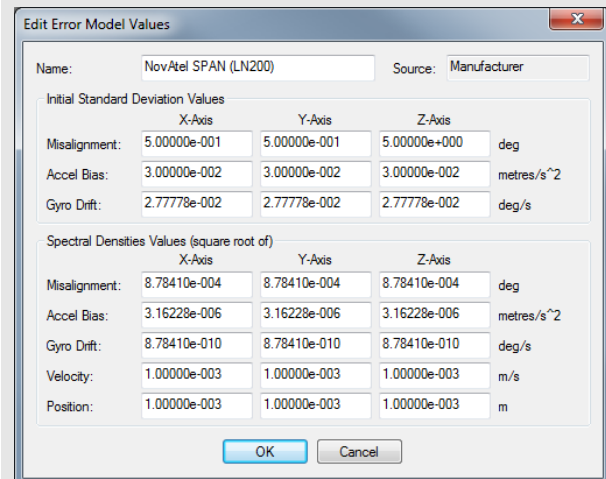
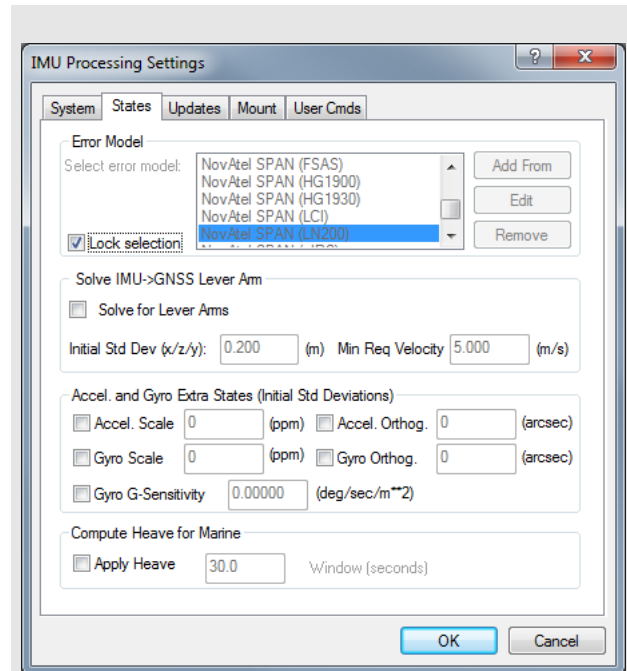
Error Model

The options available are to use one of the error models provided or use a custom model. The former option entails selecting one of the models from the scrolling window, each of which has a set of initial standard deviation and spectral density values associated with it. These models have been optimized for the indicated application or sensor through the processing of multiple sets of applicable data. Because all values have been empirically derived through a trial-and-error process, the accuracy achievable by their use cannot be guaranteed.

Certain profiles have been tailored for certain conditions, as opposed to specific sensors. For example, the *Navigation (High Precision)* model is most suitable for high-end units with gyro drift rates on the order of 0.01°/hr or better, whereas the *Tactical (Medium Precision)* model is aimed at medium-grade systems with drift rates in the range of 1°/hr to 10°/hr. The *Automotive (Low Precision)* model allows the Kalman filter to loosen up in order to compensate for poor field conditions, such as short or non-existent static alignment.

Edit Error Model Values

This is one of the most important components of processing IMU data successfully. In general, the less expensive the sensor package, the more tuning that is required to find initial variances and spectral densities that work optimally for that particular IMU. For some MEMS sensors, it is critical that spectral densities match the sensor noise of the system. In some cases, IMU processing cannot be performed at all without a properly chosen error model.



How to create a custom profile:

Click the *Add From* button to create of a new model based on a default model, or click the *Edit* button which accesses the default models. Generally, the former method is recommended, because it allows the default models, as provided by the manufacturer, to remain available for future use.

- ✉ In either case, disabling the *Lock selection* setting is required before the selection of any error model can be made. Once an error model has been selected, this setting should be re-enabled to ensure that it is not accidentally changed.

Edit Error Model Values

Name: Source:

Initial Standard Deviation Values

	X-Axis	Y-Axis	Z-Axis	
Misalignment:	5.00000e-001	5.00000e-001	5.00000e+000	deg
Accel Bias:	3.00000e-002	3.00000e-002	3.00000e-002	metres/s ²
Gyro Drift:	2.77778e-002	2.77778e-002	2.77778e-002	deg/s

Spectral Densities Values (square root of)

	X-Axis	Y-Axis	Z-Axis	
Misalignment:	8.78410e-004	8.78410e-004	8.78410e-004	deg
Accel Bias:	3.16228e-006	3.16228e-006	3.16228e-006	metres/s ²
Gyro Drift:	8.78410e-010	8.78410e-010	8.78410e-010	deg/s
Velocity:	1.00000e-003	1.00000e-003	1.00000e-003	m/s
Position:	1.00000e-003	1.00000e-003	1.00000e-003	m

Initial Standard Deviation Values

The following mathematical quantities are available:

Misalignment

These terms pertain to the difference between the computed direction cosine matrix and relate the IMU body frame to the computation frame (ECEF in Inertial Explorer) and an error-free idealized direction cosine matrix or attitude matrix.

These values represent the best estimate of the sensor's ability to compute roll, pitch and yaw during coarse alignment, assuming any is present. Roll and pitch are estimated from the sensed gravity components in the horizontal axes of the accelerometer triad, while yaw is estimated from the sensed Earth rate about the leveled gyro axes.

Typically, while even MEMS sensors can make some estimation of roll and pitch, the Earth rate is often masked by noise for lower-grade IMUs. The values entered here for x and y , which represent pitch and roll, are often as much as an order of magnitude smaller than that for z , which represents yaw.

If in doubt, simply enter large values in the range of thousand of arc seconds and allow the IMU Kalman filter time to eventually compute more sensible estimates of the error in the computation of the attitude matrix. These values must be entered in degrees.

Accel Bias

These values represent the initial uncertainties in the a priori knowledge of the constant bias errors in the accelerometer triad. If these bias values were left at zero, meaning that they are unknown, then the standard deviation values entered here should reflect this uncertainty. The processor then computes the biases on-the-fly. These values should be entered in m/s^2 .

Gyro Drift

These values refer to the initial uncertainty of the a priori knowledge of the sensor drift in the gyroscopes. If the biases are left at zero, then enter standard deviations values here that reflect this. The program attempts to compute reasonable values during processing. All values should be entered in degrees / sec.

Spectral Densities Values

Generally speaking, the lower the grade of the sensor, the larger the spectral densities that should be used for processing. As previously discussed, the spectral densities add noise to the covariance propagation process prior to filtering. Therefore, the higher the densities, the greater the weight that is placed on the GNSS updates during filtering. The following mathematical quantities are available:

Misalignment

A misalignment noise density, in degrees, becomes a covariance when multiplied by some time interval, δt . If the sensor triad is problematic in terms of providing an accurate attitude matrix, or if initial alignment is poor, then you may need to introduce large spectral density values here. These spectral components add noise to the computed Kalman covariances for misalignment, which, in turn, forces the processor to rely more heavily on the GNSS position and velocity updates. As a result, large errors in the direction cosine matrix are compensated for.

Accel Bias

Accelerometer bias densities, when multiplied by the prediction time interval, act as additive noise to the accelerometer bias states. As such, larger values here may help to compensate for large biases in the accelerometers.

Gyro Drift

Gyroscope drift densities similarly act as additives to the covariances computed for the gyroscope drift states. In the case of inexpensive units, larger values here may be necessary.

Velocity

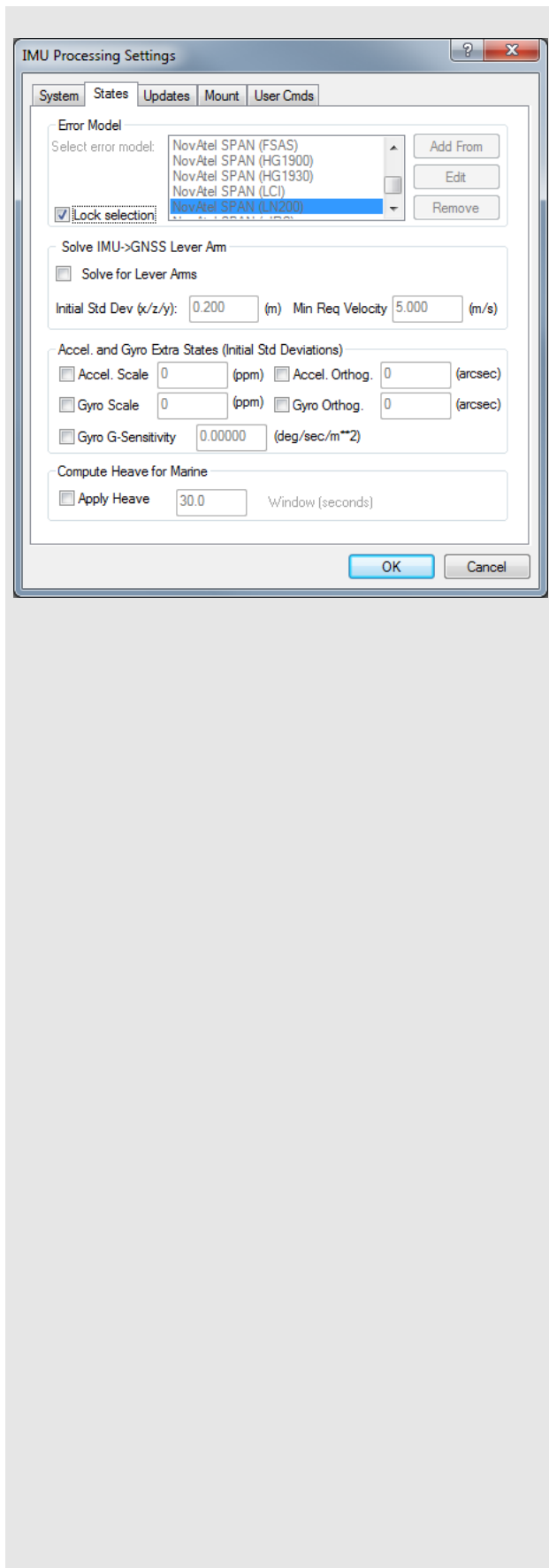
Velocity spectral densities are noise densities that account for unmodeled velocity effects during each Kalman prediction. Increasing this value permits more emphasis to be placed on the GNSS update data, but may also lead to an increase in error growth during outages. For this reason, these values should be determined as part of the tuning process. The default values are recommended unless dealing with a trajectory of unusually high dynamics, such as a race car, in which case these may need to be reduced by an order of magnitude.

Position

Position spectral densities are noise densities that account for unmodeled position effects during each Kalman prediction. Apply all of the considerations mentioned above for the velocity spectral densities.

Initial Standard Deviation Values			
	X-Axis	Y-Axis	Z-Axis
Misalignment:	5.00000e-001	5.00000e-001	5.00000e+000 deg
Accel Bias:	3.00000e-002	3.00000e-002	3.00000e-002 metres/s ²
Gyro Drift:	2.77778e-002	2.77778e-002	2.77778e-002 deg/s

Spectral Densities Values (square root of)			
	X-Axis	Y-Axis	Z-Axis
Misalignment:	8.78410e-004	8.78410e-004	8.78410e-004 deg
Accel Bias:	3.16228e-006	3.16228e-006	3.16228e-006 metres/s ²
Gyro Drift:	8.78410e-010	8.78410e-010	8.78410e-010 deg/s
Velocity:	1.00000e-003	1.00000e-003	1.00000e-003 m/s
Position:	1.00000e-003	1.00000e-003	1.00000e-003 m



Solve IMU ->GNSS Lever Arm

Lever arm accuracy is critical to post-processing results as position, velocity, and phase updates measured at the GNSS antenna phase center need to be applied at the IMU center of navigation. The position variance should be greatly increased (values of 50 to 100) if processing without accurately measured lever arms, or if solving for them using the *Solve lever arm values as additional Kalman filter states* option

Solve for Lever Arms

Use this option if the 3-D offset from the IMU sensor array center of navigation to the GNSS antenna phase center is not precisely known, or not known at all. The accuracy achieved with this option depends on the type of IMU used, but is normally better than 20 cm.

Initial Std

This value reflects the uncertainty in the entered lever arms values. For the first iteration, this value should be left at 5 meters. In any subsequent iteration, it should be lowered until the values have converged reasonably.

Minimum Velocity

This value specifies the minimum velocity that must be reached before the solving routine will start accepting data.

Accel. and Gyro Extra States

These options add scale and/or non-orthogonality states to the Kalman filter for the accelerometer and gyroscope measurements.

Apply Heave Correction to Height

For marine users who wish to apply heave compensation to the computed ellipsoidal height, use this option to utilize Inertial Explorer's low-pass filter. The algorithm requires that a window size reflecting the period of the wave motion be entered.

The value computed with this option replaces the ellipsoidal height. All trajectory files, output files and plots where ellipsoidal heights are provided with this option will now show the new heave compensated height values.

Updates

Variance Factors Applied in GNSS Residual Testing

Inertial Explorer performs residual testing using a standard least squares approach on position, phase and range updates. Phase updates, applied where there is a minimum of two satellites, are only available in tightly-coupled processing. GNSS position updates and ZUPTs, which are detected from raw IMU measurements, can be applied in either loosely- or tightly-coupled processing.

Updates are accepted only if the computed residual is within the set tolerance. The IMU processing profiles contain pre-defined position variances for NovAtel SPAN models. These have been tested to help ensure good results on typical surveys provided that the lever arm has been accurately measured.

Automated ZUPT Detection Tolerances

These settings control the software's ability to detect periods of zero velocity.

Raw Measurement

The raw gyro measurement threshold. This value may need to be raised for lower-grade sensors (i.e. MEMS) to accommodate the noisier measurements.

Velocity

The GPS velocity threshold. Potential ZUPTs are rejected if the GNSS-derived velocity exceeds this value.

Period

Length of time span over which measurements are averaged.

GNSS Updates

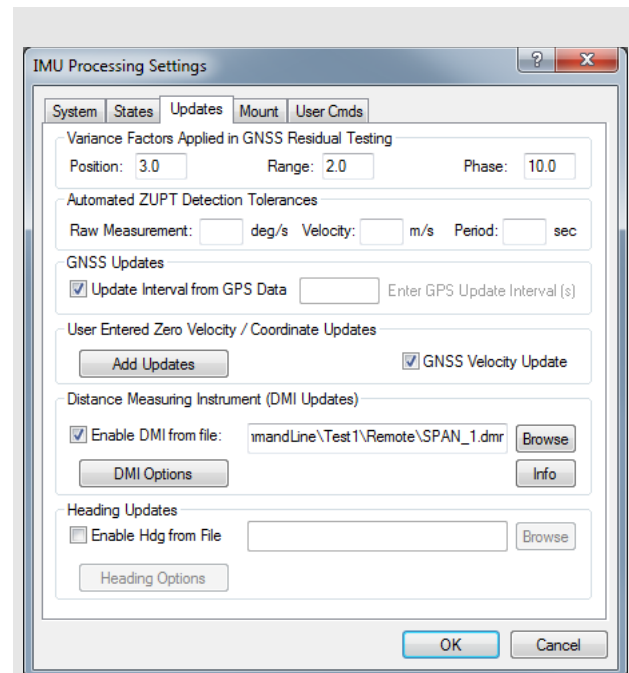
These options are listed in the shaded box.

Zero Velocity/Coordinate Updates

This option is only available for performing IMU processing without the aid of GNSS data. Otherwise, this functionality must be carried out automatically via GNSS updates throughout the mission.

GrafNav uses Doppler information to compute GNSS-derived velocities.

-
- ☒ Doppler accuracies vary significantly depending on the receiver. View the *L1 Doppler Residual RMS* values from *Output | Plot Results*. The standard deviation of the Doppler measurements used by the GNSS Kalman filter can be controlled via *Settings | Individual | Measurements*.
-



GNSS Updates Options

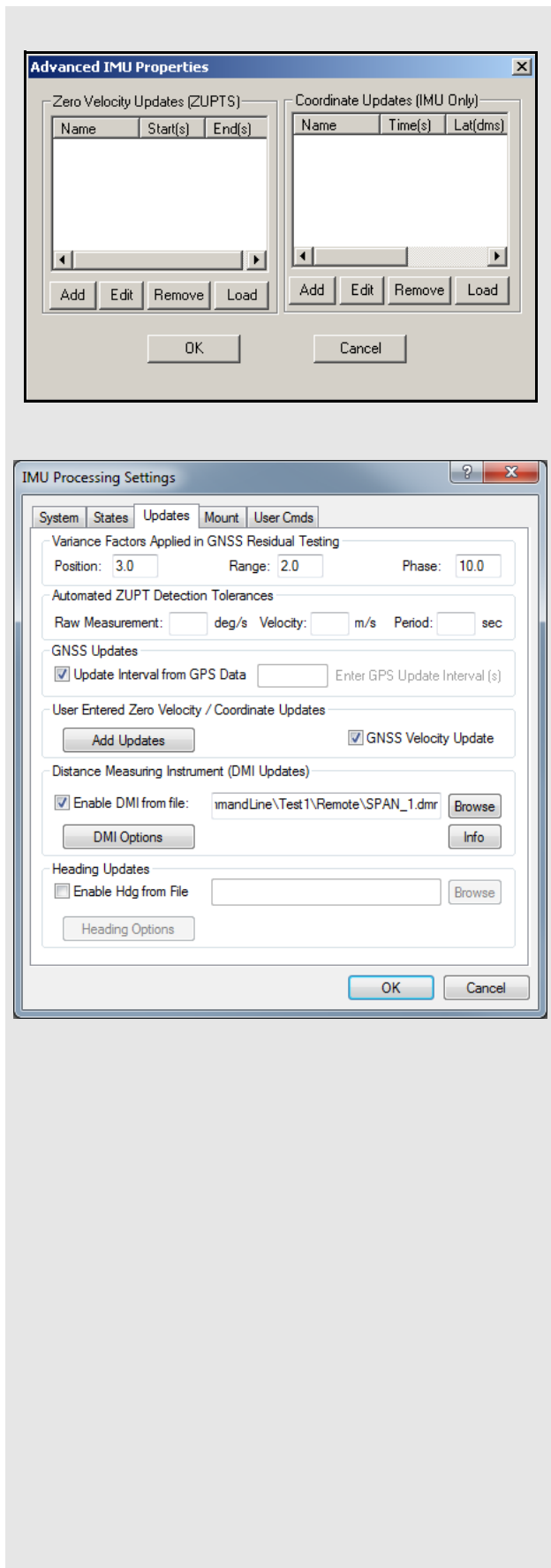
Update Interval from GPS data

This option forces the processor to use every computed epoch outputted during GNSS processing. It sets the update interval equal to that used for GNSS processing. Clear this option if you wish to manually specify the rate at which the software will perform GNSS updates. Note that the updates cannot be applied at a higher rate than the GNSS data interval

Enter GNSS update interval (s)

This value specifies the rate that Inertial Explorer performs GNSS updates. Updates cannot be performed at a rate higher than the GNSS data processing.

-
- ☒ Performing updates at a rate higher than 1 Hz is generally not helpful unless the data was collected in an environment with high dynamics.
-



Zero Velocity Updates (ZUPTS)

Inertial Explorer takes advantage of periods when the IMU is static. Such periods of time are referred to as zero velocity updates (ZUPT). Upon reaching the start of a ZUPT, Inertial Explorer assumes the data to be static and replaces the GNSS velocity update values with zeros.

- ☒ Inertial Explorer automatically detects the presence of ZUPTS by analyzing the GNSS, IMU and, if available, DMI data. This is true for both loosely and tightly coupled processing. As such, the manual entry of ZUPTS is generally not necessary, except in cases of poor data quality. Consult the *IMU Status Flag* plot after processing to determine the periods where a ZUPT was detected. If a known ZUPT was missed, manually enter it here.

Coordinate Updates (IMU Only)

Without the presence of GNSS data, the positions being computed from the inertial data are likely to drift drastically over time. This option bounds these errors by allowing the processor to correct its trajectory at any given time where the position is known. This requires that the IMU has traveled over a previously surveyed point, and that you know the exact time at which this occurred. Since it is unlikely that the IMU was physically placed upon this point, the lever arm offset should also be entered.

Distance Measuring Instrument (DMI)

To integrate the data from distance measurement instruments (DMI), select the *Enable DMI from file* option.

Enable DMI from file

Use the *Browse* button to locate the DMR file containing the measurements from the DMI. After a valid file has been selected, the software scans it to detect how many sensors were used. Use the *Info* button to view information concerning the selected file.

DMI Options

The following settings pertain to the added DMR file.

- ☒ A typical DMI will either output a tick count or a velocity vector. If tick counts are recorded, Inertial Explorer converts them into velocity vectors. If velocity vectors have been recorded, then the software uses them as such.

Detect ZUPTS from DMI sensor

A DMI can often be used to determine periods of zero velocity, which can help improve accuracies. The performance of this feature is dependent on the type

of DMI being used so it is suggested to process data with and without this option to determine its usefulness.

Measurement standard deviations

The standard deviation associated with the DMI measurements depends on the DMI being used. As such, this value may need to be determined empirically.

Wheel circumference

The default value is 1.96m. Change this value if it is not correct. Small errors are compensated for by the computed scale factor during processing. It is also possible to allow the software to determine this value based on the header in the DMR file, assuming such information is available.

Heading Updates

External heading updates are obtained by first processing the relative vector between two antennas mounted on the same vehicle as the IMU. For best results, knowledge of the accuracy of the heading update, as well as the constant offset between the IMU and the two GNSS antenna must be known.

Heading Updates from File

Certain applications benefit greatly from the use of external heading updates. Examples of these applications include where a low quality IMU is used, where it is impossible to follow recommended alignment procedures, or where a vehicle moves very slowly and/or does not experience many changes in heading.

The HMR data format is described in *Section 3.2.2, DMR File* on page 40.

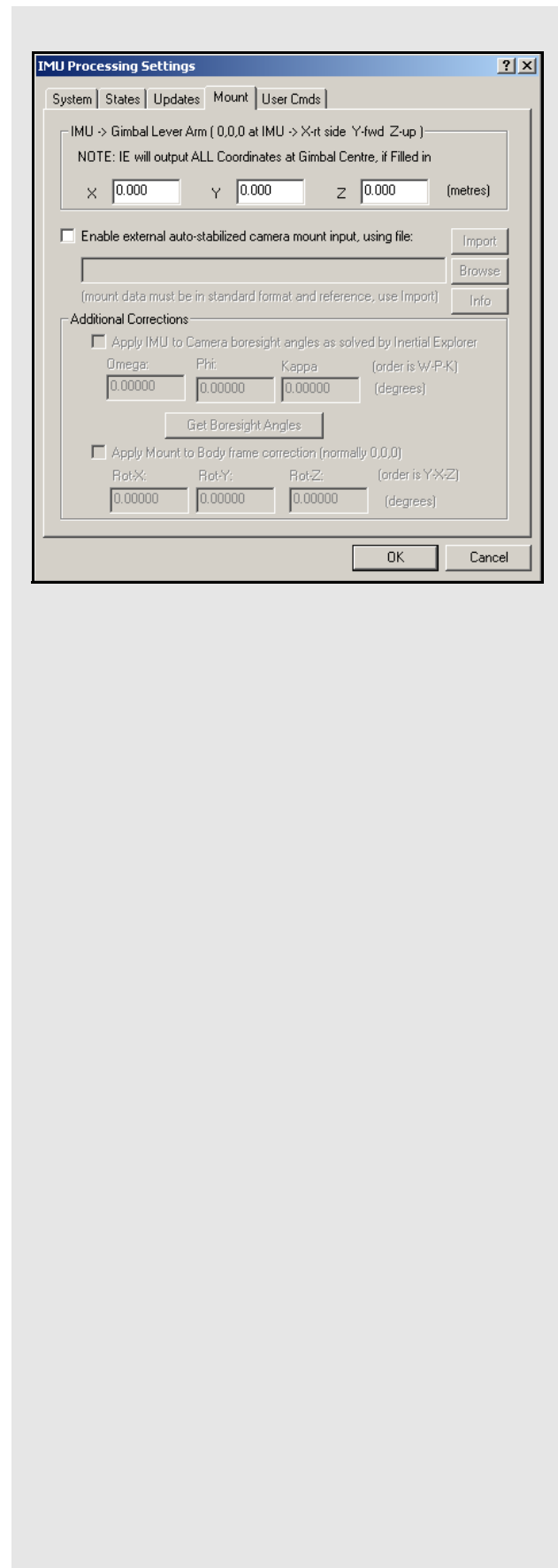
Mount

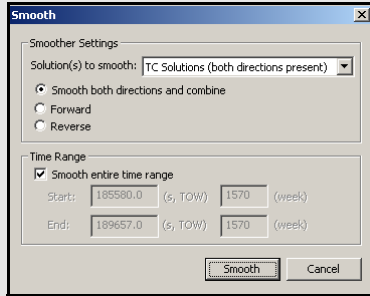
If you have mounted your IMU onto a gyro-stabilized platform, you will need to provide Inertial Explorer with the angular rotations to compensate for the changing lever arm between the IMU and GNSS antenna.

User Commands

Certain options available in Inertial Explorer cannot be accessed through the options menu. A full list of configured options is available under this tab and may be invoked to employ special features or overwrite defaults.

For an explanation of all the commands available, see *Appendix A, Summary of Commands* on page 45.





1.5.2 Combine Solutions

Refer to the *GrafNav/GrafNet 8.50 User Guide* for information regarding these options. Only points relevant exclusively to Inertial Explorer are made here.

Smooth Solutions

Inertial Explorer is capable of combining processing directions and/or performing Kalman filter smoothing on inertial trajectory. This option requires that the inertial data is already processed.

Smoothing provides the most significant results on those data sets where GNSS outages were a problem. By using this smoother, the position, velocity and attitude errors can be reduced when GNSS updates are not available. Smoothing can also clean up position and velocity accuracies even if there are no gaps.

Smoother Settings

RTS smoothing can be performed in just one direction, or both. Much like GNSS and GNSS-IMU processing, it is recommended that smoothing be performed in both directions.

Solution(s) to smooth

This option performs RTS smoothing on the loosely coupled or tightly coupled solution.

Time Range

This setting controls which period of time to perform the combining and/or smoothing of the trajectories. Epochs outside of this time range are not considered and do not appear in the output files.

1.5.3 Solve Boresighting Angles

Show

This drop-down menu is linked to the window below it and gives viewing access to the values listed in the shaded box.

Settings

The following features are available:

Calibration name

Enter a name to distinguish calibration runs from one another. Inertial Explorer keeps a history of calibration runs, so a unique identifier is helpful when trying to recover previous results. This is useful for using multiple systems and/or tracking stability over time.

Boresight Angles

Upon successful completion of the calibration procedure, the final values for the computed boresight angles are displayed here.

Add results to list

When this option is enabled, the last values computed by the program are stored so that they are easily accessible by the *Export Wizard*.

View report after computation

Enabling this option forces the software to launch the boresighting report upon successful completion of a calibration. The contents of the report are discussed later on.

Update navigation angles on entry

When this option is enabled, Inertial Explorer loads the latest navigation values for the camera events into the boresighting module.

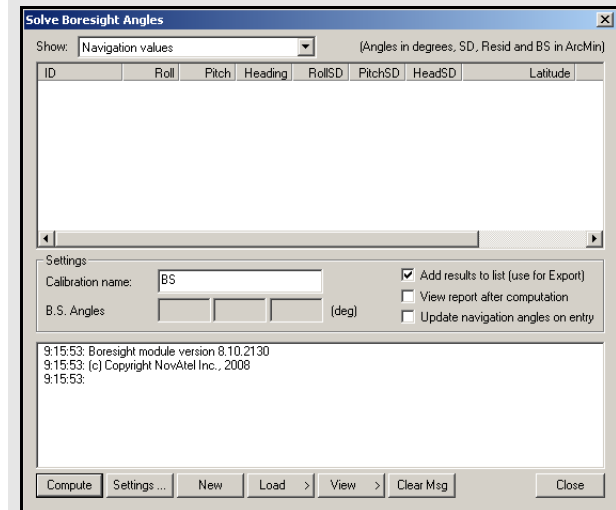
Message Window

This window provides valuable insight on the status of the current calibration. Whenever input data is being loaded, read the messages to ensure the expected number of camera events have been read in. After the calibration procedure is complete, the final boresighting values, as well as the number of iterations needed to arrive at them, are displayed.

The following options are available via the buttons along the bottom of the *Solve Boresight Angles* window:

Compute

Assuming all the required input data has been loaded, click this button to begin the iterative least squares procedure. The *Message Window* contains pertinent information regarding the success or failure of the procedure.



Values that are visible with the Show drop down menu:

Navigation values

The roll, pitch and heading values, along with their associated standard deviations, are displayed for each loaded camera event. The coordinates of the IMU at the time of the event are also displayed. These values are generally transferred from Inertial Explorer directly and correspond to the IMU values interpolated at camera event times.

Photo E/O values

The omega, phi and kappa values, along with their associated standard deviations, are displayed for each loaded camera event. These values are produced externally in a photogrammetric package.

Matches/residuals

Before the computations begin, choose whether or not to include the observations associated with a camera event in the least squares procedure by right-clicking on the event. After the least squares procedure has finished, the window is updated with the final residual values at each camera event. Additional information, such as quality indicators and computed omega, phi and kappa values are also displayed.

Axes/System Definition options

System

The selection made here defines the ground coordinate system to which the omega, phi and kappa values are oriented. Normally, they are referenced to a map projection which is defined in the *Grid/Map Definition* settings.

Order

This setting defines the order in which the omega, phi and kappa angles are to be applied during the transformation from the ground system to the image or IMU system. Only the omega-primary, phi-secondary and kappa-tertiary rotation order is supported.

Axes

Use this setting to define the orientation of the image system. The most commonly used system is the conventional frame, where the x-axis points forward, the y-axis points left, and the z-axis points upwards. The frame defined here determines the composition of the R_c matrix.

Settings...

This button gives access to the *Boresight Settings* window, which is useful for configuring many parameters used in the boresight calibration.

Axes/System Definition

These options are listed in the shaded box.

Grid/Map Definition

The options made available here depend on the system definition chosen. If the input angle was provided with respect to a map grid, then the selection made here determines the convergence value, α , used to form the R_g matrix. In addition, grid users are given the opportunity to enter the average ground height in order to maximize accuracy.

Measurement Weighting

The selections made here determine the composition of the variance-covariance matrix used in the least squares procedure to derive the final boresighting values. Choose to enter a set of constant standard deviation values to apply to all measurements, or have the values derived from either the navigation SD values, the photo SD values (if provided), or a combination of both.

The other setting here pertains to the outlier tolerance. The value specified here determines at which point a measurement is removed from the least squares procedure.

Display Units

These options pertain to the values displayed in the *Solve Boresight Angle* window and determine which units are used when writing to the *Boresight Report* file. These options also allow the number of decimal places to which all values are displayed or written to be modified.

New

This button clears any stored data from previous calibration runs in order to start a new one.

Load

Use this button to load the required navigation and exterior orientation input data.

The navigation data can be obtained either by loading the latest set of roll, pitch and heading values computed by Inertial Explorer, or by an external file which contains this information for each camera event. Alternatively, if such information is available, there is the ability to provide the module directly with the omega, phi and kappa angles required to rotate the ground system into the IMU frame. Obtaining the attitude angles directly from Inertial Explorer is by far the most common usage.

The exterior orientation parameters for each photo must be supplied by an external file. This file should contain the omega, phi and kappa angles required to rotate the ground system into the image system.

View

This button gives access to the post-calibration report. The report contains relevant boresight calibration information, as well as a list of all the input data provided for each camera event. The bottom of the report displays the boresight values and residuals from the final iteration.

- ☒ This report can be viewed through either NotePad or the internal Inertial Explorer ASCII viewer. This button also gives you access to the calibration history. For each calibration run, the final boresighting results are saved, assuming the *Add results to list* option is enabled.

Clear Msg

This button clears the *Message Window* of any messages currently displayed.

1.6 Settings Menu

Refer to the *GrafNav/GrafNet 8.50 User Guide* for information regarding all of the features available from this menu.

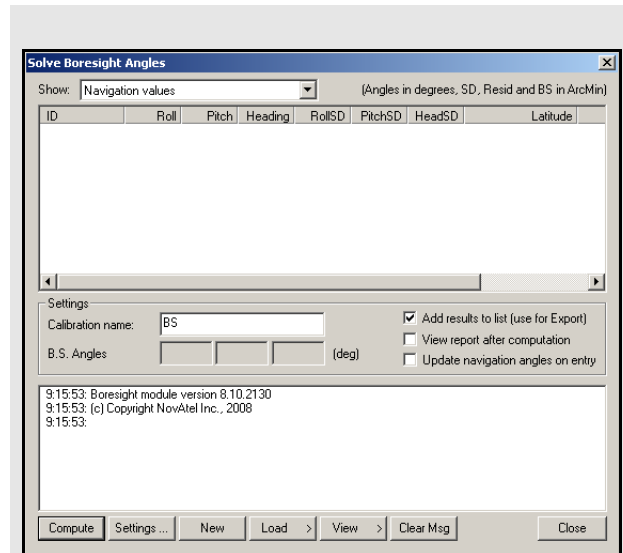


Table 1: IMU Plots

Plot	Description
Accelerometer Bias	This is the apparent output in acceleration when there is no input acceleration present. It is computed by the GNSS/INS Kalman filter and the effects may be sinusoidal or random. It is plotted in terms of the X (right direction), Y (forward direction), and Z (up direction) of the INS body. Generally, they should stabilize after the alignment period and agree when processed in both directions.
Attitude (Azimuth/Heading)	Plots the heading and GNSS COG (course-over-ground) that was computed from the GNSS/INS processing. Effects of crabbing is visible in this plot if the GNSS COG bears a constant offset from INS heading. The <i>IMU Heading COG Difference</i> plot shows the difference between these two heading values. Note that any transitions between a heading of 359 degrees and 0 degrees shows up as a vertical line.
Attitude (Roll and Pitch)	Plots the roll and pitch values from GNSS/INS processing. In airborne data, it is common to see roll values between 30 degrees and pitch values of around 10 degrees, depending on the flight pattern of the aircraft itself.
Attitude Separation	This plot shows the difference between the forward and reverse solutions in terms of roll, pitch and heading. A zero separation is ideal, as it indicates matching solutions in the forward and reverse IMU processing. Spikes at the beginning and the end of the plot are common, as they indicate the periods of alignment.
Body Frame Acceleration	This plot shows the components of acceleration in the vehicle body frame.
Body Frame Velocity	This plot shows the components of velocity in the vehicle body frame.
DMI Scale Factor	This plot presents the DMI scale factor, as computed by the Kalman filter. It should be loaded separately for forward and reverse processing to ensure that the same scale factor is computed in both directions. Ideally, the plotted line should be horizontal, indicating a constant scale factor.
DMI Residual	This plot presents the difference between the computed displacement or velocity and that reported by the DMI.
DMI Analysis Tool	This tool allows DMI users to view the raw data measurements found in their DMR file. They can use the options available here to find an appropriate scale factor that will make the DMI data fit best with the values computed from the GNSS-IMU data.
Estimated Accelerometer Bias Accuracy	This plot shows the estimated standard deviation of the accelerometer bias. It is plotted in terms of the X (right direction), Y (forward direction), and Z (up direction) of the INS body.
Estimated Attitude Accuracy	This plot shows the standard deviation computed in the GNSS/INS Kalman filter in terms of roll, pitch and heading.

(Sheet 1 of 3)

1.7 Output Menu

Refer to the *GrafNav/GrafNet 8.50 User Guide* for information regarding all of the features available from this menu. Only those features exclusive to Inertial Explorer are discussed here.

1.7.1 Plot Results

Refer to the *GrafNav/GrafNet 8.50 User Guide* for information regarding all of the GNSS plots available.

By default, the software generates all plots at the GPS update interval. You can raise the interval as high as the IMU data rate to get a denser plot, but generation takes longer. This setting is available under the X-axis tab. This option requires that you have generated a combined binary file for your trajectory (before or after smoothing) (see *Section 1.5.3, Solve Boresighting Angles* on page 27). The combined file must be re-generated after every processing run to ensure that the plot reflects the latest results.

Table 1, IMU Plots, in the shaded box on this page and the following page, contains descriptions of the IMU plots available only through Inertial Explorer.

1.7.2 Export Wizard

Only the *Export Wizard* window exclusive to Inertial Explorer is discussed here. Refer to the *GrafNav/GrafNet 8.50 User Guide* for additional information concerning this feature.

IMU Epoch Settings

Epoch/Output Interval

The *Kalman/GPS interval* indicates the interval used during GPS processing, while the *IMU integration interval* displays the interval at which the IMU data was processed. These values can only be changed prior to processing. The *Output data interval* defines the interval to export solutions. The output interval can be set as high as 1000 Hz, regardless of what interval the data was processed at.

The time range for which to obtain outputs can also be limited here.

Lever Arm/ Offset

Allows for the coordinates of the IMU, calculated via the IMU Kalman filter, to be transferred to an alternate sensor's location.

Note the orientation of the frame in which these coordinates must be entered. You are also free to save your offset for future use via the *Favorites* button.

1.8 Tools Menu

Refer to the *GrafNav/GrafNet 8.50 User Guide* for information regarding all of the options available via this menu.

1.9 Interactive Windows

Refer to the *GrafNav/GrafNet 8.50 User Guide* for information regarding the *Map Window* and the features available within it.

1.10 Processing Window

Table 1, IMU Plots on page 30, contains a list of the additional parameters available for viewing in Inertial Explorer during processing. Display these values via the *View* button in the *Processing Window*.

The values in the *GrafNav/GrafNet 8.50 User Guide* differ in the manner in which they are computed depending on the mode of processing being performed.

If the GNSS is being processed, then the values displayed are those computed in the Kalman filter. However, during the IMU processing, the values displayed reflect those calculated in the IMU Kalman filter, using the GNSS information as updates. Ideally, these values should agree. When they do not, monitor the position and velocity misclosure.

1.11 Help Menu

1.11.1 Help Topics

Opens an HTML version of this manual, with the GrafNav portion included. This feature is a quick and easily accessible reference.

1.11.2 NovAtel Waypoint Products Web Page

This option opens a web browser to the Waypoint Products Group’s page on the NovAtel website. On this page are found details on the latest versions, patches, information on GNSS/INS, and technical reports.

Table 1: IMU Plots (continued)

Plot	Description
Estimated Gyro Drift Accuracy	This plot shows the estimated standard deviation of the gyro drift rate, which generally decreases with time. It is plotted in terms of the X (right direction), Y (forward direction), and Z (up direction) of the INS body.
Gimbal Data Values	Inertial Explorer supports data processing from IMU's mounted on a gyro-stabilized platform (gimbal). When doing so, it is required to import an MMR file in order to compensate for the changing lever arm between the IMU and GPS when the gimbal is engaged. This plot shows the decoded rotations within the MMR file and can thus be useful in troubleshooting if the values in the MMR file are suspect.
Gyro Drift Rate	This is the apparent change in angular rate over a period of time, as computed by the GNSS/INS Kalman filter. The effects are usually random. It is plotted in terms of the X (right direction), Y (forward direction), and Z (up direction) of the INS body. Generally, they should stabilize after the alignment period and agree when processed in both directions.
Gyro Attitude Misclosure	This plots shows the misclosure (residual) of gyroscope Kalman filter updates. Large values here could be an indication of attitude instability.
IMU Angular Rates	This plot shows the gyroscope rate of change of attitude in the X, Y and Z axes of the IMU body with the drift removed. This plot is used to check the gyros.
IMU Status Flag	Shows the status of IMU processing. Specifically, this plot provides indication of the type of update, if any, being applied at each epoch.
IMU-GPS Lever Arm	This plots presents the body-frame components of the lever arm offset between the IMU and GNSS antenna. If the offset was manually entered, then this plot has constant horizontal lines. If left to be solved by the Kalman filter, this plot shows the computed values.
IMU Heading COG difference	This plot is the difference between the IMU heading and the GNSS course-over-ground values. Effects of crabbing shows up as a direct bias in this plot.
Velocity Separation	Plots the difference between the East, North and Up components of velocity computed during forward and reverse processing. Requires that both directions be processed and combined.
IMU-GPS Position Misclosure	This plot shows the difference between the GNSS solution and the mechanized INS positions obtained from the GNSS/INS processing. This is a good analysis tool used to check the GNSS/INS solution as well as checking INS stability. Large jumps or spikes may indicate a bad INS solution, whereas separations nearing zero confirms the GPS solution.

(Sheet 2 of 3)

Table 1: IMU Plots (continued)

Plot	Description
IMU-GPS Velocity Misclosure	This plot shows the difference between the GNSS calculated velocity and the mechanized INS velocity obtained from the GNSS/INS processing. Another good analysis tool used to check INS stability.
IMU Heading COG difference	This plot is the difference between the IMU heading and the GNSS course-over-ground values. Effects of crabbing shows up as a direct bias in this plot.
Raw IMU Data Values	Use this plot to see the raw gyroscope and accelerometer measurements as they appear in the IMR file.

(Sheet 3 of 3)

Manufacturer files available to download from Waypoint's FTP site

manufact.dn1

List of base stations available for the *Download* utility. This is usually updated monthly.

manufact.dtm

List of datums, ellipsoids, and transformations between datums. Refer to the *GrafNav/GrafNet 8.50 User Guide* for details.

manufact.fvt

List of *Favourites* and the groups they are contained in.

manufact.grd

List that contains available grids such as UTM, US State Plane, Gauss Kruger etc. Refer to the *GrafNav/GrafNet 8.50 User Guide* for more details.

manufact.svi

A file that associates a PRN number with a satellite type (Block II, Block IIA etc.) for purposes of determining the center of mass of the satellite. Used by PPP.

manufact.cim

Provides a set of default conversion parameters for various IMU raw data formats. This file is accessed during the conversion from raw data to IMR format. See *Chapter 2, Conversion Utilities* on page 33 for details.

manufact.imu

Contains the error profile parameters for various types of IMUs. The values associated with each model have been empirically derived by the Waypoint Products Group staff based solely on the data sets they have encountered.

manufact.dcb

List of the differential code biases, in nanoseconds, between the P1 and C/A code for each satellite. Used by PPP.

1.12 About Inertial Explorer

This window displays information about the software version, build dates, copyright information, hardware lock key information and DLL information.

Access the hardware key utility from this window by clicking *Key Util (Upgrade)*. This tool is useful for upgrades. The *Dependent Files* window displays a list of executables and DLLs associated with Inertial Explorer. The date and time of the files are shown, as well as a quick description of the file.

Click *Download latest manufacturer files* to connect to Waypoint's FTP site (Internet connection required) to download the files listed in the shaded box.

Most of these *manufact* files have an associated user file, where personal information is saved. These files are not modified when you download the latest manufacturer files. It is important, for this reason, to not modify the *manufact* files, as they are overwritten when this option is used.

2.1 Raw IMU Data Converter

The *IMU Data Converter* utility is a Win32 application program that converts custom data formats into a generic raw IMU data format. This utility is available exclusively to users of Inertial Explorer and may be accessed from *File | Convert | Raw IMU Data to Waypoint Generic (IMR)*.

2.1.1 Waypoint IMU Data Conversion

Input/Output Files

Refers to the names and locations of all input and output files.

Input Binary IMU File

Click the *Browse* button to locate the raw IMU data file.

Output Waypoint Binary File

By default, the binary output file created is given the same filename as the input file, but with an IMR extension. It is saved to the directory containing the input file.

Output Waypoint ASCII File

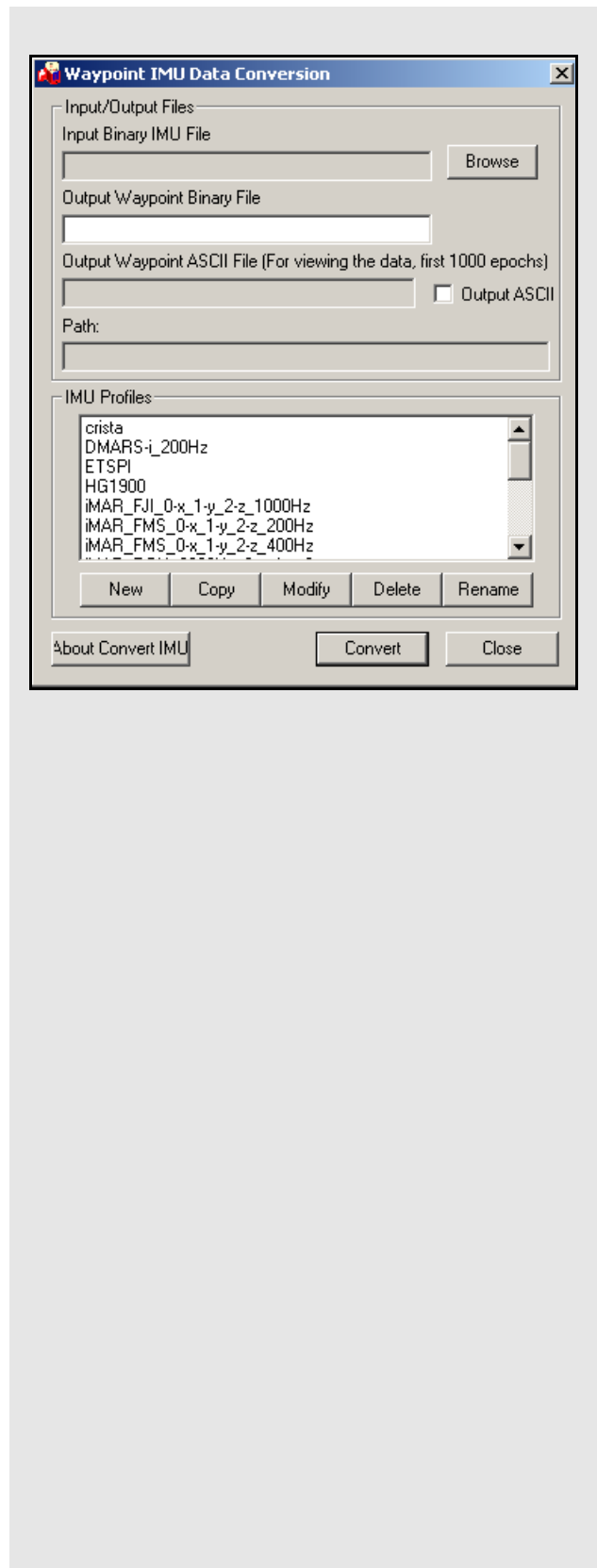
If the *Output ASCII* option is enabled, the utility generates an ASCII file containing the GPS time, as well as the gyroscope and acceleration measurements of all three axes for the first thousand epochs. Use this to detect any errors that may occur during the conversion, such as the use of an incorrect scale factor.

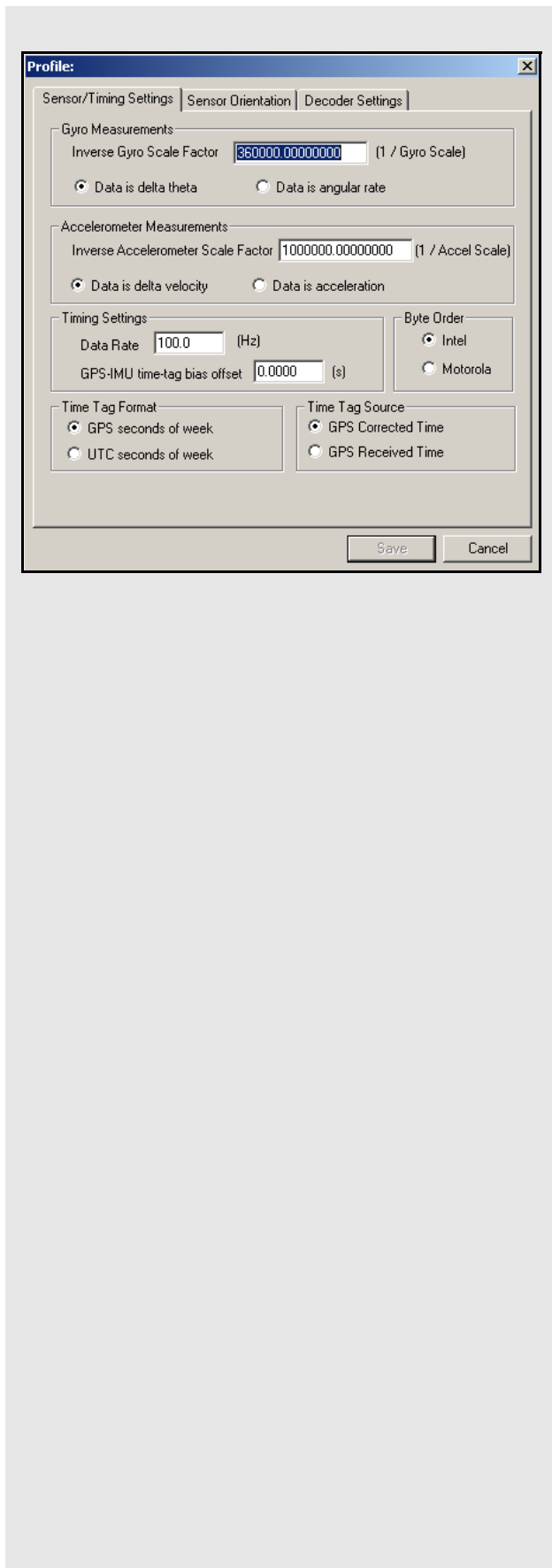
Path

Displays the path to the directory containing the input file. All output files created by this utility are saved to this directory.

IMU Profiles

Displays a scroll-down list of profiles available for use during conversion. Each profile contains a set of conversion parameters designed to decode measurement data files produced by the indicated sensor. Choose one profile from the list, or, if necessary, create one. See *Section 2.1.2, Creating / Modifying a Conversion Profile* on page 34 for help. After all the appropriate fields have been entered, click the *Convert* button to start converting IMU data into IMR format. A message window appears to show the status of the conversion process.





2.1.2 Creating / Modifying a Conversion Profile

New

Creates a customized profile to convert a unique format into Waypoint's generic IMR format. This is used for custom scale factors, data rates, and orientations in raw data files.

Modify

Allows changes to be made to an existing profile.

Delete

Deletes an existing profile.

Rename

Renames an existing profile.

Sensor/Timing Settings

Gyro Measurements

Pertains to the measurements made by the gyroscopes.

-
- ☒ The inverse value of the scale factor is required. For example, a scale factor of 0.0004, which can be represented fractionally by 1/2500, should be entered as 2500.
-

The gyro measurements can take the form of *delta theta*, where angular increments are being observed, or *angular rate*.

Accelerometers Measurements

Similar to the scale factor of the gyro measurements, the inverse of the accelerometer scale factor is required. As well, the accelerometer measurements can take two forms, the first being *data velocity*, or *velocity increments*, and the other being *acceleration*.

Timing Settings

Enter the data collection rate of the IMU sensor and specify any offset that may exist between the GNSS and the IMU time tags.

Byte Order

This flag must be properly set.

-
- ☒ If the proper byte order is not specified, the decoding of the binary raw file will fail.
-

Time Tag Format

There are two options available as the data is either acquired in the GPS time frame or the UTC time frame. This must be correctly identified in order for the IMU data to be properly aligned with the GNSS data.

Time Tag Source

Specify whether the time tags represent the *GPS Corrected Time* or the *GPS Received Time*.

2.1.3 Sensor Orientation Settings

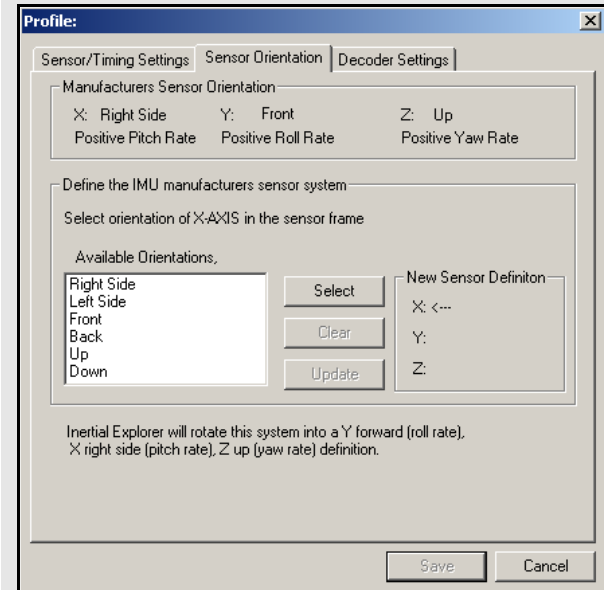
Define the orientation of the IMU here using the steps in the shaded box.

The orientation will always be right-handed.

2.1.4 Decoder Settings

Specifies which library is used to perform the conversion, based on the input format of the raw data file. For most sensors, this should be left untouched.

For SPAN, the IMU decoding is handled through the GNSS decoder.



How to define the orientation of the IMU:

1. Specify the X-direction by selecting the direction that corresponds to the X-axis of the sensor frame.
2. Click *Select* to set that direction to the X-axis.
3. Specify the Y-direction by selecting the direction that corresponds to the Y-axis of the sensor frame.
4. Click *Select* to set that direction to the Y-axis.

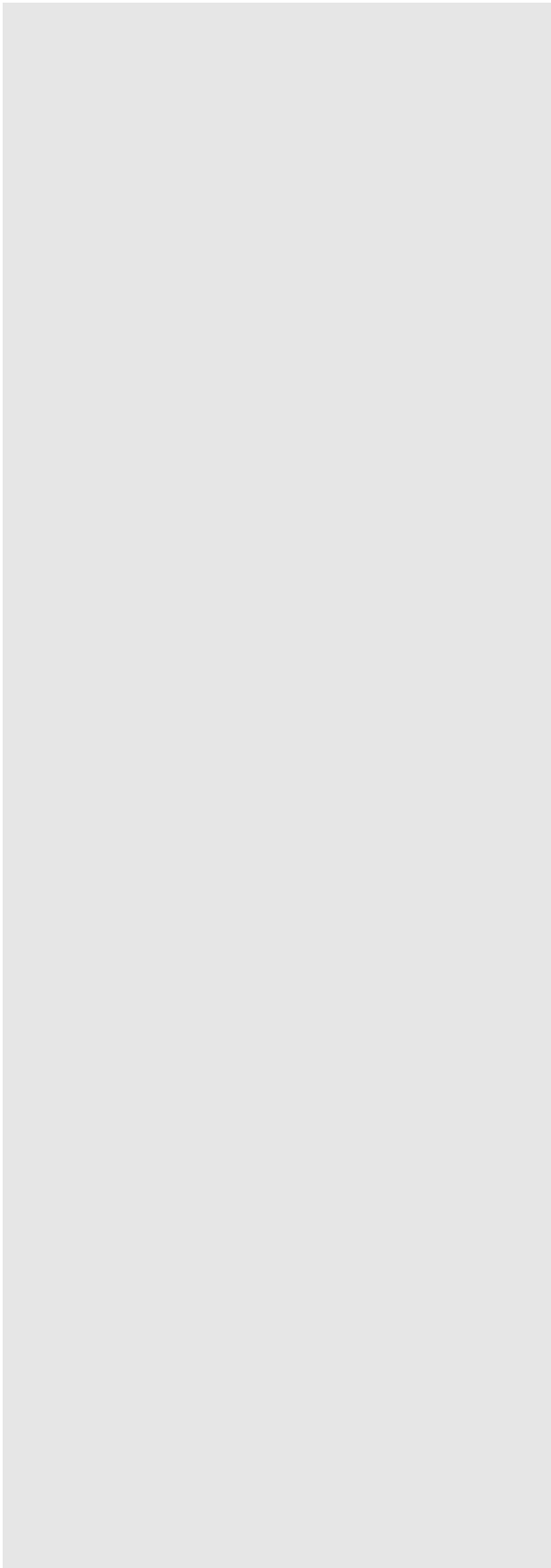
Given the constraint that the frame is right-handed, this direction will be automatically determined by the software.

5. Click *Update* to apply the new sensor orientation to the profile.

If a mistake is made at any point during the process, click *Clear* to start over.

6. Click *Save* to save the new profile.

It should immediately appear in the scroll-down list under the *IMU Profiles* box of the main window.



3.1 Data Formats

In theory, virtually any IMU sensor can be used with Inertial Explorer. The only requirement is that the data be logged in the format provided in this section, which allows easy decoding with the *IMU Data Conversion* utility described in *Section 2.1.1, Waypoint IMU Data Conversion* on page 33.

Table 2, Binary Structure of Raw Data presents the binary structure in which the conversion utility expects the raw IMU data to be logged.

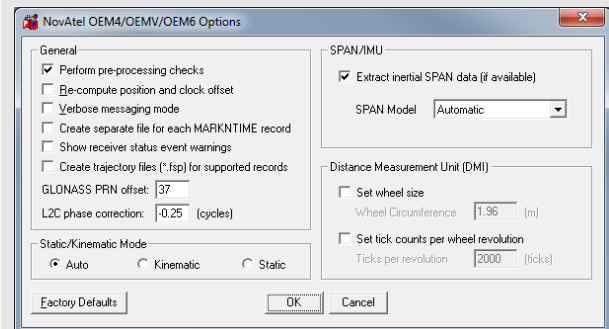
3.1.1 NovAtel's SPAN Technology

With the use of NovAtel's SPAN technology, note the difference in the IMU data decoding procedure. Since the raw IMU data measurements are embedded into the same binary file containing the raw GNSS measurements, only one step is needed to separate the data and convert it into the Waypoint Group's format. Therefore, the *Raw IMU Data Converter* utility does not need to be used.

Instead, decode the GNSS and IMU data simultaneously via the *Convert Raw GNSS data to GPB* utility, which can be accessed from *File | Convert*. When adding the measurement file to the *Convert Files* window for decoding, ensure that the drop-down menu under the *Receiver Type* box has been set to *NovAtel OEM4/OEMV/OEM6*. Then, click either the *Global Options* or *Options* button to gain access to the IMU decoding settings.

Table 2: Binary Structure of Raw Data

Word	Size (bytes)	Type	Description
GpsTime	8	real	time of the current IMU rate measurements in GPS seconds of the week
GyroX	4	long	scaled X-body axis gyro measurement as an angular increment or angular rate
GyroY	4	long	scaled Y-body axis gyro measurement as an angular increment or angular rate
GyroZ	4	long	scaled Z-body axis gyro measurement as an angular increment or angular rate
AccelX	4	long	scaled X-body axis accelerometer measurement as a velocity increment or acceleration
AccelY	4	long	scaled Y-body axis accelerometer measurement as a velocity increment or acceleration
AccelZ	4	long	scaled Z-body axis accelerometer measurement as a velocity increment or acceleration



✉ AG11 = AG58
AG17 = AG62

3.2 File Formats

3.2.1 IMR File

Waypoint converts all custom IMU raw binary formats into a generic format (IMR), which is read from Inertial Explorer following the decoding process in *IMU Data Converter*. See *Chapter 2, Conversion Utilities* on page 33 for more details.

Because it contains vital information for reading and decoding the data, the first 512 bytes of the generic IMU data format is a header which must be filled in, read and interpreted. In a C/C++ structure definition, the generic format header has the following fields:

```
struct imr_header_type
{
    char    szHeader[8];           // $IMURAW[0] - NULL terminated ASCII string

    char    bIsIntelOrMotorola;   // 0 – Intel (Little Endian) - default
                                   // 1 – Motorola (Big Endian) - swap bytes for IExplorer
                                   // This can be set for any user who directly writes in our
                                   // format with a Big Endian processor. IExplorer will swap the bytes

    double  dVersionNumber;      // Program version number (i.e. 8.50)

    int     bDeltaTheta;         // Default is 1, which indicates the data to follow will be delta
                                   // thetas, meaning angular increments (i.e. scale and divide by
                                   // by dDataRateHz to get degrees/second). If the flag is set to 0, then
                                   // the data will be read directly as scaled angular rates

    int     bDeltaVelocity;      // Default is 1, which indicates the data to follow will be delta v's,
                                   // meaning velocity increments (i.e. scale and divide by
                                   // dDataRateHz to get m/s2). If the flag is set to 0, then the data will
                                   // be read directly as scaled accelerations

    double  dDataRateHz;         // i.e. 100.0 records/second. If you do not know it, set this to zero
                                   // and then fill it in from the interface dialog boxes

    double  dGyroScaleFactor;    // Scale (multiply) the gyro measurements by this to get degrees/sec,
                                   // if bDeltaTheta=0. Scale the gyros by this to get degrees, if
                                   // bDeltaTheta =1. If you do not know it, then the data can not be
                                   // processed. Our default is to store the gyro data in 0.01 arcsec
                                   // increments or 0.01 arcsec/sec, so that GYRO_SCALE = 360000

    double  dAccelScaleFactor;   // Scale (multiply) the accel measurements by this to get m/s2
                                   // if bDeltaVelocity=0. Scale the accels by this to get m/s, if
                                   // bDeltaVelocity =1. If you do not know it, the data can not be
                                   // processed. Our default is to store the accel data in 1e-6 m/s
                                   // increments or 1e-6 m/s2, so that ACCEL_SCALE = 1000000

    int     iUtcOrGpsTime;       // Defines the time-tags as being in UTC or GPS seconds of the week
                                   // 0 – Unknown (default is GPS), 1 – UTC, 2 – GPS

    int     iRcvTimeOrCorrTime;  // Defines whether the GPS time-tags are on the nominal top of the
                                   // second or are corrected for receiver time bias
                                   // 0 – do not know (default is corrected time)
                                   // 1 – receive time on the nominal top of the epoch
                                   // 2 – corrected time i.e. corr_time = rev_time - revr_clock_bias
}
```

```

double dTimeTagBias;           // default is 0.0, but if you have a known millisecond-level bias in
                               // your GPS→INS time tags, then enter it here

char szImuName[32];           // Name or type of inertial unit that is being used

unsigned char reserved1[4];    // Reserved for future use; bytes should be zeroed

char szProgramName[32];       // Name of calling program; skip if writing directly to this format

time_type tCreate;           // Creation time; skip if writing directly to this format (12 bytes)

bool bLeverArmValid;          // Set to true if the sensor definition that follows is valid
                               // Lever arm is from IMU to GPS phase centre

long lXoffset;                // X value of lever arm, in millimeters
long lYoffset;                // Y value of lever arm, in millimeters
long lZoffset;                // Z value of lever arm, in millimeters

char Reserved[354];           // Reserved for future use; bytes should be zeroed
};

```

The single header, which is a total of 512 bytes long, is followed by a structure of the following type for each IMU measurement epoch:

```

typedef struct
{
    double Time;                // GPS time frame – seconds of the week
    long gx,gy,gz;              // delta theta or angular rate depending on flag in the header
    long ax,ay,az;              // delta v or acceleration depending on flag in the header
} INS_type;                     // this is the binary structure type expected in GPSIMU

```

-
- ☒ The angular increments (or angular rates) are scaled long integers. The scale factor to obtain a double precision word must be supplied in the header. Similarly, the accelerations (or velocity increments) are signed four byte words and must be scaled by a double precision variable given in the header.
-

3.2.2 DMR File

☒ All odometer data must be written into Waypoint's generic format (DMR) before it can be used within Inertial Explorer.

```

struct dmi_hdr_type
{
    char szHdr[8];                // $DMIRAW[0] - NULL terminated ASCII string

    short sHdrSize;              // Size of header, in bytes. Must be set to 512

    short sRecSize;              // Size of each record (refer to dmi_lrec_type and dmi_drec_type)
                                // 12 + 8*sDim if sValueType = DMI_VALUE_DOUBLE
                                // 12 + 4*sDim if sValueType = DMI_VALUE_LONG
                                // where sDim is number of DMI sensors

    short sValueType;           // Number type (DMI_VALUE_LONG/DOUBLE)
                                // 0 if logging data using LONG values
                                // 1 if logging data using DOUBLE precision

    short sMeasType;            // Measurement type (distance or speed)
                                // 1 if logging a distance measurement
                                // 2 if logging a speed measurement

    short sDim;                 // Number of DMI sensors
                                // Maximum is 3, but only 1 can be used in Inertial Explorer

    short sRes;                 // Measurement resolution of DMI
                                // 1 if low resolution (i.e. only makes measurements on the full wheel revolution)
                                // 3 if high resolution (i.e. makes measurements at partials of a wheel revolution)
                                // or on fixed time intervals

    short sDistanceType;        // Type of distance measurement
                                // Must be set if sMeasType = 1 (distance measurements)
                                // 1 if logging accumulated tick count
                                // 2 if logging distance, in meters
                                // 3 if logging accumulated distance, in meters

    short sVelocityType;        // Type of velocity measurement
                                // Must be set if sMeasType = 2 (velocity measurements)
                                // 1 if logging velocity in meters/second
                                // 2 if logging velocity in ticks/second

    double dScale;              // Scale factor (m/count or m/s/count)
                                // Must be set if sValueType is set to 0
                                // 1.0 if logging accumulated tick count or ticks/seconds
                                // If logging in meters or meters/second, then dScale will
                                // scale measurements into corresponding units

    char szAxisName[DMI_MAX_DIM][16]; // Name of various axes/DMI; optional; NULL terminated

    double dWheelSize;          // Size of the wheel, circumference in meters
                                // Must be set if logging accumulated tick count or ticks/second

```



```

long lTicksPerRevolution;    // Number of tick counts per wheel revolution
                             // Must be set if logging accumulated tick count or ticks/second

char cExtra2[420];         // Reserved for future use; bytes should be zeroed
};

```

The single header, which is a total of 512 bytes, is followed by one of the following structure types for each DMI measurement record:

```

struct dmi_lrec_type        // If logging using LONG values
{
    short sSync;           // Sync byte
                           // Set to 0xffee

    short sWeek;          // GPS week number; set to -1 if not known

    double dTime;         // GPS time of week, in seconds

    unsigned long lValue[DMI_NUM_DIM];
                           // values (counts)
                           // DMI_NUM_DIM should be equal to sDim
};

struct dmi_drec_type        // If logging using DOUBLE precision
{
    short sSync;           // Sync byte
                           // Set to 0xffee

    short sWeek;          // GPS week number; set to -1 if not known

    double dTime;         // GPS time of week, in seconds

    double dValue[DMI_NUM_DIM]; // values (double precision)
                           // DMI_NUM_DIM should be equal to sDim
};

```

3.2.3 HMR File

The 256 byte header contains information that is vital to processing and must be filled in. The C/C++ structure definition of the HMR header is as follows:

```
typedef struct
{
    char szTitleStr[12];           // $SIMUHEADING[0]; NULL terminated ASCII string
    unsigned char ucType;         // Set to 1 if external or 2 if dual antenna
    double dBoreSightRotationZ;   // Heading boresight rotation about Z, in degrees
                                   // Set to zero if unknown
                                   // Use positive boresight rotation as clockwise from north
                                   // IE will use the yaw definition by negating this so we have a
                                   // right-hand definition that fits with internal computations
    double dBoreSightRotationZStdDev;
                                   // accuracy of the boresight, in degrees; zero if unknown
    char Extra[227];             // Reserved; bytes should be zeroed
} heading_hdr_type;             // 256 bytes
```

The single header is then followed by the 34-byte structure type below for each heading update record:

```
typedef struct
{
    double dGpsTime;             // GPS time of week, in seconds
    short sGpsWeek;             // GPS week number; set to -1 if unknown
    double dHeading;            // Heading update, in decimal degrees
                                   // Use positive rotation clockwise from north
                                   // IE uses yaw (i.e. rotation counterclockwise from north)
                                   // yaw = -heading
    float fHeadingStdDev;       // Standard deviation of update (decimal degrees)
                                   // zero if unknown
    float fBaselineLength;      // Distance between antennas, in metres
                                   // Only if ucType = 2 (dual antenna)
    float fPitch;               // Pitch between two antennas, in degrees
                                   // Only if ucType = 2 (dual antenna)
    float fPitchStdDev;         // Standard deviation of the pitch, in degrees; zero if unknown
} heading_rec_type;           // 34 bytes
```

3.3 Output Files

This section discusses the different output files that are created when processing with Inertial Explorer.

3.3.1 FIL/RIL/FTL/RTL Files

Message Log files echo all error and warning messages sent to the *Process Window* during INS processing. Some example messages are below.

Coarse Alignment Succeeded on attempt #1

This message appears at the beginning of forward/reverse processing. It indicates that the coarse alignment was successful.

Reading GrafNav Record: 1000

Scanning the GPS/IMU data before processing.

Warning: Coarse Alignment has used up all of the Align Time. Will Continue Processing anyway...

This message indicates that the INS processor used all the coarse alignment time entered.

Warning - GUPT Missing in GPS data -> DT: 0.200

This message indicates the presence of a gap in GPS data of time DT seconds. As a result, the processor cannot use this time as a GPS update.

**Warning: Trace of Body-Earth matrix < 0 after update
Check on Fix: 1-tr{C} of transform matrix (0.056389)**

This message is purely a mathematical computation warning.

3.3.2 FIM/RIM/FTM/RTM Files

These files contain the trajectory information computed by the inertial filter and are available for both tightly-coupled and loosely-coupled processing. They are typically output at a minimum rate of 1Hz but can be higher depending on the GNSS update interval.

The first line of the output file always begins with *\$OUTREC*, and is followed by the version number, the processing engine, and the type of output. An example is given below:

```
$OUTREC Ver8.30.0329 GPSINS DLL Forward GpsInsOutput
```

The format of these ASCII trajectory files is outlined within the header of the forward/reverse files and are not discussed here.

3.3.3 BIF/BIR/BTF/BTR Files

Here is the C/C++ structure definition of the header, which is 512 bytes:

```
typedef struct
{
  char Str[16];           // $IMUOUT
  long HdrSize;          // size of this header
  long IsExtended;      // true if extended format used
  long RecSize;          // size of this record (see note below)
  long Reserved1;        // for later, zero
  double Interval;      // data interval (s)
  char ProgramName[32];  // program name that created this file
  char VersionName[32];  // version that produced file
  char Direction[16];    // "Forward", "Reverse" or "Combined"
  char Reserved2[392];   // reserved for future use (zero at creation of new file)
} imu_hdr_type;
```

☒ The RecSize variable should always be checked as the structure size and format may change in future versions. Contact Waypoint Support if you have any questions.

The single header is then followed by the 121-byte structure type below for each processed epoch:

```
typedef struct
{
  double GpsTime;        // GPS time of this record (seconds of week)
  short WeekNum;         // week number
  unsigned char FixedFlagAndHeave[2];
                          // bit 0: Fixed solution=on/float solution=off
                          // bits 1-14: heave (+/-16.384 m)
                          // bit 15: sign of heave
  plh_type_double GeoPos; // geographic position of this record (deg, deg, m)
  fxyz_type LLVel;        // local level velocity (m/s)
  fxyz_type LLAcc;        // local level acceleration (m/s2)
  iatt_type LLAtt;        // local level attitude (deg) (scaled)
  fxyz_type LLAttDot;     // body frame rotational rate (deg/s)
  fxyz_type stdPos;       // position standard deviations (m)
  fxyz_type stdVel;       // velocity standard deviations (m/s)
  fxyz_type stdAtt;       // attitude standard deviations (deg)
  char Reserved[1];
} imu_outrec_type;

typedef struct
{
  double phi, lamda;     // latitude and longitude (deg)
  double ht;             // ellipsoidal height (m)
} plh_type_double;

typedef struct
{
  float x, y, z;
} fxyz_type;

typedef struct
{
  signed long iRoll, iPitch, iYaw; // attitude (deg); scaled by 1.0e-6
} iatt_type;
```

The rotation matrix from body-to-local level is defined as: $R_B^{LL} = R_3(\text{yaw}) * R_1(\text{pitch}) * R_2(\text{roll})$

ACCEL_BIASES = x y z	Accelerometer biases, in m/sec
ACCEL_NOISE = sd1 sd2 sd3	Spectral densities of the accelerometer bias states, in m^2/s^5
ACCEL_SD = x y z	A priori Kalman standard deviations for accelerometer bias states, in m/s^2
ALIGNMENT_MODE = mode	[0]coarse/fine align, [1]coarse align only, [2]no alignment (transfer or on-the-fly alignment)
ASCII_OUTPUT_INTERVAL = Interval	ASCII IMU output to FIM/RIM files. Default is 1 Hz
COARSE_ALIGNMENT_MODE = mode	[0]static coarse alignment, [1]enter the az, [2]transfer alignment
COARSE_ALIGNMENT_TIME = time1 time2	Seconds of coarse alignment for FWD/REV processing
CORRELATION_TIMES = gyro_time accel_time	Correlation time for Gauss-Markov gyroscope and accelerometer bias states, in seconds
DATA_RATE = rate	Data rate for IMU: 64 Hz for LTN90 50 Hz for LRF3 50 Hz for LN200 100 Hz for HG1700
DYNAMICS_MODE = mode	[0]low, [1]high, [2]extreme
	ERROR_MODEL_IMU = Name Error model name from <i>manufact.imu</i> or <i>user.imu</i>
GPS_COVARIANCE_SCALE = pos vel	Scale GPS Kalman co-variances for positions and velocities by this amount
GPS_HEADING_UPDATES = OFF/ON const StdDev	ON/OFF - Use GPS heading updates / constant offset (deg) / standard deviation of offset angle (deg)
GPS_INS_OFFSETS = dx dy dz	Lever arm, in metres, measured in body frame
GPS_INS_UPDATE_TOL = FLOAT/FIXED sdev Q D_rms	Only use GPS updates if standard deviation, quality factor, and Doppler RMS are below tolerances
GPS_POSITION_UPDATES = ON/OFF	Enable/disable GPS position updates
GPS_TRAJECTORY_SOURCE = source	Set GPS solution file ‡ [0]CMB, [1]FWD, [2]REV, [3]external file
GPS_VELOCITY_UPDATES = ON/OFF	Enable/disable GPS velocity updates
GPS_UPDT_INT = n	GPS updates every <i>n</i> seconds FIXED/FLOAT (ambiguities ‡ def FLOAT) std_dev_GPS_coord (def ‡ 2m) Q-factor (1-6 ‡ def 6) DopplerRms(m/s)
GRAFNAV_FILE = filename1 filename2	Name of trajectory source file for GPS updates (forward and reverse)
GYRO_BIASES = x y z	X,Y,Z gyro drifts, in deg/sec
GYRO_NOISE = sd1 sd2 sd3	Spectral density of the gyro drift states, in $arcsec^2/sec^3$

GYRO_SD = xsdev ysdev zsdev	A priori Kalman standard deviations for X,Y,Z gyro drift states, in arcsec/sec
INITIAL_ATTITUDE_KNOWN = att	ATTITUDE_KNOWN, HEADING_KNOWN, ATTITUDE_UNKNOWN. Refers to validity of the initial roll, pitch, yaw values.
INITIAL_ATTITUDES = rowfwd pitchfwd azfwd rowrev pitchrev azrev	If known, input initial roll, pitch, azimuth (degrees) for forward and reverse alignment
INITIAL_INS_POSITION = latfwd lonfwd htfwd latrev lonrev htrev	INS position for forward and reverse processing, in DMS, DMS, m
INITIAL_POSITION_SOURCE = source	[0]CMB/FWD/REV, [1]manual entry of position
INITIAL_VELOCITIES = vefwd vnfwd vhfwd verev vnrev vhrev	If known, input initial East, North, Up velocities (m/s) for forward and reverse alignment
INS_BODYFRAME_OFFSETS = roll pitch yaw	Constant angular offsets, in degrees, from the IMU axes to vehicle body frame axes
INS_DATAGAP_TOL = Num_consecutive_epochs_missing	Exit on a single data gap that is larger than this number
INS_FILE_NAME = filename	Name of IMU binary measurement file
INS_GPS_TIME_FRAME = GPS_TIME_FRAME/RCVR_TIME_FRAME	GPS_TIME_FRAME (i.e. rcvr_time - rcvr_clock_bias)/RCVR_TIME_FRAME (i.e. nominal rcv_time)
INS_LOWPASS = 0/1	Use 1 for low-pass filter on raw data
INS_PROCESS_DIR = FORWARD/REVERSE	FORWARD/REVERSE direction INS processing
INS_PROCESS_MODE = mode	GPS_INS or INS_ONLY
INS_TIMERANGE = start end	Indicates IMU processing time range, and whether to process ALL or PARTIAL
INS_TYPE = type	LTN90, LRF-3, LN200, HG1700, JAMI_IMU, GENERIC_IMU(*.imr)
INS_VERBOSE = ON/OFF	Writes extra messages to FIL and RIL files
MISALIGNMENT_NOISE = sd1 sd2 sd3	Spectral density of misalignment states, in arcsec^2/s
MISALIGNMENT_SD = esddev nsddev hsddev	A priori Kalman standard deviations for East, North, Up misalignment states, in arc seconds
NUM_STATES = number	Number of Kalman filter states
POSITION_MEAS_VAR = cov	Variance (R-matrix) of Kalman CUPT observation, in m^2
POSITION_NOISE = sd1 sd2 sd3	Spectral density of the coordinate states, in m^2/s
POSITION_SD = esdev nsdev usdev	A priori Kalman standard deviations for E-N-U coordinate states, in metres
PREDICTION_RATE = n	Kalman filter predictions every n seconds. Default is 0.5.
TIME_OFFSET = offset	Correction, in seconds, to GPS times in the IMU binary file. Default is zero.

TOTAL_ALIGNMENT_TIME = time1 time2	Seconds of coarse + fine alignment. Note that no fine alignment is performed when total_align_time = coarse_align_time
VEL_MEAS_VAR = var	Variance (R-matrix) of Kalman ZUPT observation, in m^2/s^2
VELOCITY_NOISE = sd1 sd2 sd3	Spectral density of the velocity states, in m^2/s^3
VELOCITY_SD = sdev1 sdev2 sdev3	A priori Kalman standard deviations for E-N-U velocity states, in metres/sec
WRITE_BINARY_OUTREC = ON/OFF	Write binary structure to disk at mechanization rate for FWD/REV combination
ZUPT_ALIGN_INTERVAL = interv	Length of time for a ZUPT during fine alignment.
ZUPT_AT = name start end	Name, start time (GPS sec) and end time (GPS sec) of ZUPT
ZUPT_OPTION = num	[0]one epoch or [1]best fit over ZUPT_length seconds. Default is 1.
ZUPT_TIME = time	Minimum length of time for a ZUPT
ZUPT_VEL_TOL = tol	Reject a ZUPT with velocity above this tolerance, in m/s

Index

A

- accelerometer, 9, 17, 20
- Add, 12
 - Master File (s), 12
 - Remote File, 12
- Alignment methods, 16
- antenna
 - frame, 15
 - height, 12
 - lever arm, 10, 15, 22, 31
- Auto Start
 - description, 12
 - how to use, 12

B

- BIF File data structure, 44
- BIR File data structure, 44
- Boresighting angles, 27
- BTF File data structure, 44
- BTR File data structure, 44

C

- coarse alignment description, 16
- combining solutions
 - forward and reverse, smoothed, 26
- Convert
 - GNSS Data, 10
 - IMU Data, 10
- Converting
 - creating profile for raw data conversion, 34
 - raw IMU data to IMR, 33

D

- data interval
 - for GPS updates in Inertial Explorer, 23
- DMI Options, 24
- DMR File, 40

E

- error model
 - settings, 19
- Export
 - Final Coordinates, 11
- export wizard, 30

F

- FIL File
 - see Message Logs, 43
- File Data Coverage
 - Determine Alignment Method, 10

FIM File

- see Trajectory File, 43

fine alignment, description, 17

FTL Files

- see Message Logs, 43

FTM File

- see Trajectory File, 43

G

GNSS updates

- Using, 23

H

Heave

- applying, 22

HMR File, 42

- format and description, 42

I

IMR File

- converting to, 33
- format and description, 38

IMU

- Process loosely coupled setting, 14

IMU file

- adding to Inertial Explorer, 12

In-Motion Kinematic Alignment

- description, 17

L

lever arm

- solve values, 22

lever arm offset

- IMU to alternate sensor, 30
- IMU to GNSS, 15
- read values, 15

Loosely coupled processing, 14

M

manufacturer files

- downloading latest files, 32

Message Logs

- format and description, 43

Mount tab, 25

N

NovAtel SPAN Data, 37

P

plots

- IMU data, 30

- list of Inertial Explorer plots, 30

Plotting, 11

- Attitude, 11

Processing IMU options, 14

Processing Profiles

- GNSS/IMU, 18

processing window, 31

Project Wizard, 12

- steps, 12

Q

quick start, 9

R

Remote File

- Adding, 12

RIL File

- see Message Logs, 43

RIM File

- see Trajectory File, 43

rotation

- sensor frame to body frame, 15

RTL File

- see Message Logs, 43

RTM File

- see Trajectory File, 43

RTS Smoother

- options, 26

S

Smoother

- see RTS Smoother, 26

System tab, 14

T

Tightly coupled processing, 25

time range

- processing range in Inertial Explorer, 18

Trajectory File

- format and description, 43

U

updates

- coordinate, 23

- GPS source file, 14

- GPS velocity, 23

- zero velocity, 24

User Defined Options

- adding commands, 25

- list of commands, 45

V

Variance factors, 23