WAYPOINT PRODUCTS GROUP

A NovAtel Precise Positioning Product

Inertial Explorer®

User Manual

Inertial Explorer 8.70 User Manual

Publication Number: OM-20000166

Revision Level: v2

Revision Date: November 2016

This manual reflects Inertial Explorer software version 8.70.

Warranty

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

Software Warranty	One (1) year

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

TOC

Chapter 2 Conversion Utilities

Chapter 3 Data and File Formats

End-User License Agreement ("EULA")

IMPORTANT NOTICE: BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE, FIRMWARE, SCRIPT FILES, UPGRADES, UPDATES OR OTHER ELECTRONIC PRODUCT WHETHER EMBEDDED IN THE HARDWARE, ON A CD OR AVAILABLE ON THE COMPANY WEB SITE DELIVERED WITH THIS EULA (HEREINAFTER COLLECTIVELY AND INDIVIDUALLY REFERRED TO AS "SOFTWARE PRODUCT"), "YOU" (EITHER AN INDIVIDUAL OR SINGLE ENTITY) AGREE TO BE BOUND BY THE TERMS OF THIS EULA, WHICH WILL TAKE PRECEDENCE OVER ANY OTHER DOCUMENT AND SHALL GOVERN USE OF THE SOFTWARE PRODUCT. IF YOU DO NOT AGREE WITH THESE TERMS OF USE, YOU ARE NOT AUTHORIZED TO DOWNLOAD, INSTALL AND COPY OR USE THIS SOFTWARE PRODUCT.

1. LICENSE

NovAtel Inc. ("NovAtel") grants you a non-exclusive, non-transferable license (not a sale) to use the SOFTWARE PRODUCT subject to the limitations below. The duration of the license grant is defined by the individual SOFTWARE PRODUCT purchased and the quotation. You agree not to use the SOFTWARE PRODUCT for any purpose other than the due exercise of the rights and licenses 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 PRODUCT and the SOFTWARE PRODUCT is protected by national copyright laws, international treaty provisions and all other applicable national laws. You must treat the SOFTWARE PRODUCT like any other copyrighted material and the SOFTWARE PRODUCT may only be used on one computer at a time. You may not copy the product manual or written materials accompanying the SOFTWARE PRODUCT. No right is conveyed by this EULA 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 EULA, without the prior written consent of NovAtel.

3. THE FOLLOWING ARE PROHIBITED FOR YOUR LICENSE:

3.1 You may not use the SOFTWARE PRODUCT on more than one computer simultaneously;

3.2 You may not distribute, transfer, rent, lease, lend, sell or sublicense all or any portion of the SOFTWARE PRODUCT without the written permission of NovAtel;

3.3 You may not attempt to disable or work around any software licensing security mechanisms that are part of the SOFTWARE PRODUCT thus disabling the software copy protection;

3.4 You may not modify or prepare derivative works of the SOFTWARE PRODUCT;

3.5 You may not use the SOFTWARE PRODUCT in connection with computer-based or cloud-based services business without the written permission of NovAtel;

3.6 You may not publicly display visual output of the SOFTWARE PRODUCT without crediting NovAtel and the SOFTWARE PRODUCT name;

3.7 You may not implement DLLs and libraries in a manner that permits automated internet based post-processing (contact NovAtel for special pricing);

3.8 You may not reverse engineer, decompile or disassemble the SOFTWARE PRODUCT; or

3.9 You may not use the SOFTWARE PRODUCT for any purposes associated with development or production of chemical, biological or nuclear weapons or their delivery systems.

4. TERM AND TERMINATION

This EULA is effective until terminated or until your software subscription or lease expires without being renewed (as the case may be). In the event that You shall, at any time during the term of this EULA be in breach of your obligations hereunder where such breach is irremediable or if capable of remedy is not remedied within thirty (30) calendar days of notice from NovAtel requiring its remedy; then and in any event NovAtel may forthwith by notice in writing terminate this EULA together with the rights and licenses hereby granted by NovAtel. You may terminate this EULA by providing written notice to NovAtel. You agree upon the earlier of the termination of this EULA or expiration of your software subscription to cease using and to permanently destroy the SOFTWARE PRODUCT (and any copies, modifications and merged portions of the SOFTWARE PRODUCT in any form, and all of the component parts of the SOFTWARE PRODUCT) and certify such destruction in writing to NovAtel. Termination shall be without prejudice to the accrued rights of either party, including payments due to NovAtel. This provision shall survive termination of this EULA howsoever arising.

5. WARRANTY

5.1 Warranty. NovAtel does not warrant the contents of the SOFTWARE PRODUCT or that it will be error free. The SOFTWARE PRODUCT is furnished "AS IS" and without warranty as to the performance or results you may obtain by using the SOFTWARE PRODUCT. The entire risk as to the results and performance of the SOFTWARE PRODUCT is assumed by you.

5.2 Disclaimer. THE WARRANTIES IN THIS EULA REPLACE ALL OTHER WARRANTIES, AND NOVATEL EXPRESSLY DISCLAIMS ANY AND ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

5.3 NovAtel will not be liable for any loss or damage caused by delay in furnishing the SOFTWARE PRODUCT or any other performance under this EULA.

6. CUSTOMER SUPPORT

6.1 Post Contractual Support (PCS). Each SOFTWARE PRODUCT license has a PCS subscription period associated with it. Perpetual SOFTWARE PRODUCT licenses have, by default, one-year of PCS subscription from the time of purchase. Thereafter, PCS subscription periods can be extended by purchasing additional periods in one-year increments. SOFTWARE PRODUCT which is licensed under a term subscription or lease shall be entitled to PCS benefits for the term of the license as defined in Section 4.

6.2 While within a PCS subscription period you are entitled to:

6.2.1 Bug fixes and maintenance patches (“Updates”) and version releases and enhancements (“Upgrades”) if and when released during the PCS subscription period for the covered SOFTWARE PRODUCT.

6.2.2 Expert phone and e-mail support

6.3 For software Updates and Upgrades (defined below), and regular customer support, contact the NovAtel Support\ Hotline at 1-800-NOVATEL (U.S. or Canada only), or 403-295-4900, Fax 403-295-4901, e-mail to support@novatel.com, website: http://www.novatel.com or write to: NovAtel Inc., 1120 - 68 Avenue NE, Calgary, Alberta, Canada T2E 8S5.

6.4 Software Version Support. NovAtel will support versions of the SOFTWARE PRODUCT for a minimum of three (3) years from the SOFTWARE PRODUCT release date. This support period includes version-specific auto-download content such as manufacturer files.

6.5 Lost licenses. You are responsible to ensure that your licenses are properly tracked and maintained. NovAtel is not responsible for any lost licenses due to lost, stolen or damaged computers, or in cases where the license cannot be repaired.

7. AUDIT

NovAtel shall have the right, during your normal business hours, to audit your use of the SOFTWARE PRODUCT and your compliance with the provisions of this EULA. NovAtel will provide you with thirty (30) days prior written notice of an audit. The right of audit shall be limited to twice per calendar year. Prior to the start of an audit, NovAtel’s personnel will sign a reasonable non-disclosure agreement provided by you. During the audit, you shall allow NovAtel’s personnel to be provided reasonable access to both your records and personnel. The cost of the audit shall be paid by NovAtel unless the results of the audit indicate that you have underpaid fees to NovAtel, in which case, you agree to promptly pay NovAtel such fees at the price previously agreed to for the SOFTWARE PRODUCT license or software subscription plus interest on such underpayments from the original due date at the lesser of two percent (2%) per month or the highest rate allowed by applicable law, and you further agree to bear all costs associated with the audit.

8. INDEMNIFICATION

NovAtel shall not be liable to indemnify You 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 PRODUCT by You or any other information supplied or to be supplied to You pursuant to the terms of this EULA. 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, You shall co-operate reasonably with NovAtel in all legal actions concerning this license of the SOFTWARE PRODUCT under this EULA taken against any third party by NovAtel to protect its rights in the SOFTWARE PRODUCT. NovAtel shall bear all reasonable costs and expenses incurred by You in the course of co-operating with NovAtel in such legal action.

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 You and You 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 your use of the SOFTWARE PRODUCT 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.

9. LIMITATION OF LIABILITY

Notwithstanding anything to the contrary, to the maximum extent permitted by applicable law, in no event shall NovAtel be liable for any indirect, incidental, consequential, punitive or special damages, including but not limited to, loss of use, loss of data, loss of business information, business interruption, loss of revenue or loss of profit even if NovAtel has knowledge of the possibility of such damages. NovAtel’s total liability under this EULA shall not exceed the amount that NovAtel was paid by You for the SOFTWARE PRODUCT under this EULA. Except as required by applicable law, no claim, regardless of form, arising out of or in connection with this EULA may be brought by You more than one (1) year after the cause of action as occurred.

10. RESTRICTIONS

10.1 United States Government Restricted Rights. If the SOFTWARE PRODUCT (including any Updates, Upgrades, documentation or technical data related to such SOFTWARE PRODUCT) is licensed, purchased, subscribed to or obtained, directly or indirectly, by or on behalf of a unit or agency of the United States Government, then this Section 10.1 also applies:

10.1.1 For civilian agencies. The SOFTWARE PRODUCT was developed at private expense and is "restricted computer software" submitted with restricted rights in accordance with the Federal Acquisition Regulations ("FAR") 52.227-19 (a) through (d) (Commercial Computer Software – Restricted Rights).

10.1.2 For units of the Department of Defense. The SOFTWARE PRODUCT was developed at private expense and is "commercial computer software" submitted with restricted rights in accordance with the Defense Federal Acquisition Regulations ("DFARS") DFARS 227.7202-3 (Rights in commercial computer software or commercial computer software documentation).

10.2 Export Restrictions. This SOFTWARE PRODUCT, including any technical data related to this SOFTWARE PRODUCT, is subject to the export control laws and regulations of the United States, Canada and the United Kingdom, including, but not limited to the U.S. Export Administrations Act. Diversion contrary to United States, Canadian and United Kingdom law is prohibited. This SOFTWARE PRODUCT, including any technical data related to this SOFTWARE PRODUCT and any derivatives of this SOFTWARE PRODUCT, shall not be exported or re-exported, directly or indirectly (including via remote access), under the following circumstances:

10.2.1 No export will be made contrary to any Canadian, U.S., or U.K. export regulation;

10.2.2 Export compliance will never be compromised for commercial gain;

10.2.3 Intentional violations of export laws and regulations will not be tolerated. If You intentionally or knowingly participate in violation of export regulations, this EULA is subject to immediate termination and possible legal action. You may also be exposed to criminal or civil litigation, fines or imprisonment; and

10.2.4 Where there is any doubt, uncertainty or suspicion of possible impropriety with respect to any law surrounding any proposed transaction, the transaction will be forfeited without question.

Canadian, U.S., and U.K. laws restrict the export or re-export of SOFTWARE PRODUCT to certain people, countries, and end users. You agree to comply with these laws and will not sell, or otherwise provide SOFTWARE PRODUCT to anyone regardless of location in violation of these export restrictions, including without limitation, the U.S. Export Administration Regulations ("EAR"), International Traffic in Arms Regulations ("ITAR"), the Arms Export Control Act ("AECA"), the Export Administration Act ("EAA"), as implemented through the International Economic Emergency Powers Act), the Anti-Boycott Regulations and Guidelines issued under the Export Administration Act, as amended, Section 999 of the Internal Revenue Code ("Anti-Boycott Regulations"), and the Foreign Corrupt Practices Act ("FCPA").

You agree to comply with all export laws, rules and regulations of the United States and their respective foreign export compliance agencies or authorities, and not to export or re-export the SOFTWARE PRODUCT in violation of any such laws, rules or regulations, or without all necessary authorizations. You have the obligation to obtain and bear all expenses relating to any necessary licenses and/or exemptions with respect to its export of the SOFTWARE PRODUCT from your country.

Any breach of the obligations or representations set forth in this Section shall be deemed to be a material breach of this EULA, entitling NovAtel to terminate this EULA without notice and seek such remedies as may be appropriate in the circumstances.

11. GENERAL

11.1 Entire Agreement. This EULA constitutes the entire agreement between the parties hereto with regard to the subject matter of the SOFTWARE PRODUCT. This EULA supersedes any and all prior discussions and/or representations, whether written or oral, and no reference to prior dealings may be used to in any way modify the expressed understandings of this EULA. Any future representations, promises and verbal agreements related to the SOFTWARE PRODUCT, including but not limited to features, future enhancements, functionality, or services covered by this EULA will be of no force or effect unless reduced in writing and made a part of this EULA.

THIS EULA MAY NOT BE AMENDED OR MODIFIED UNLESS SO DONE IN WRITING SIGNED BY AUTHORIZED REPRESENTATIVE OF NOVATEL. THE PRE-PRINTED TERMS AND CONDITIONS OF ANY PURCHASE ORDER OR ANY OTHER TERMS AND CONDITIONS OF A PURCHASE ORDER WHICH MAY CONFLICT IN ANY WAY WITH THE TERMS AND CONDITIONS OF THIS EULA SHALL BE VOID, EVEN IF ISSUED SUBSEQUENT TO THE EFFECTIVE DATE OF THIS EULA, AND SHALL NOT BE DEEMED TO CONSTITUTE A CHANGE TO THIS EULA.

11.2 Severability. If a provision of this EULA is or becomes or is found by a court or other competent authority to be illegal, invalid or unenforceable, in whole or in part, under any law, such provision will to that extent only be deemed not to form part of this EULA and the legality, validity and enforceability of the remainder of this EULA will not be affected or impaired. The parties will negotiate in good faith to replace any such illegal, invalid or unenforceable provision with a valid and enforceable provision which, as far as possible, has the same legal and commercial effect as that which it replaces.

11.3 No Waiver. No delay or failure on the part of any party in exercising a right, power or remedy provided by law or under this EULA will impair that right, power or remedy or operate as a waiver of it or any other rights and remedies. The single or partial exercise of any right, power or remedy provided by applicable mandatory law or under this EULA will not preclude any other or further exercise or the exercise of such rights, power or remedy.

11.4 Governing Law and Venue. This EULA shall be interpreted under the laws of the Province of Alberta, Canada. This EULA shall not be governed by the conflict of law rules of any jurisdiction or the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded. In the event of a dispute arising out of or relating to this EULA, the parties agree that venue is proper in and that they will submit irrevocably to the exclusive jurisdiction of the courts of relevant jurisdiction in Calgary, Alberta, Canada.

11.5 Notices. Any notice or other communication (“Notice”) required or permitted under this EULA shall be in writing and either delivered personally or sent by electronic mail, facsimile, overnight delivery, express mail, or certified or registered mail, postage prepaid, return receipt requested. A Notice delivered personally shall be deemed given only if acknowledged in writing by the person to whom it is given. A Notice sent by electronic mail or facsimile shall be deemed given when transmitted, provided that the sender obtains written confirmation from the recipient that the transmission was received. A Notice sent by overnight delivery or express mail shall be deemed given twenty-four (24) hours after having been sent. A Notice that is sent by certified mail or registered mail shall be deemed given forty-eight (48) hours after it is mailed.

If any time period in this EULA commences upon the delivery of Notice to any one or more parties, the time period shall commence only when all of the required Notices have been deemed given. NovAtel’s address for Notices is NovAtel Inc., 1120 – 68th Avenue N.E., Calgary, Alberta T2E 8S5 Canada, Attn: Contracts Department, +1-403-295-4500.

11.6 Assignment. Neither Party shall assign any of its rights or delegate any of its obligations under this EULA without the prior written consent of the other party, provided that such consent shall not be unreasonably withheld, except that NovAtel may assign its rights and obligations under this EULA without your to an entity which acquires all or substantially all of the assets of NovAtel Inc. or to any subsidiary, affiliate or successor in a merger or acquisition of NovAtel Inc.

Foreword

Congratulations!

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 *Customer Service* below 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.

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 GrafNav / GrafNet GNSS data post-processing software. The conventions include the following:

> This is a note box that contains important information before you use a command or log, or to give additional information after wards.

Customer Service

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

Call: 1-800-NovAtel (1-800-668-2835) for North American access
1-403-295-4900 for International access

Email: support@novatel.com

Web: www.novatel.com/support/

Chapter 1 Inertial Explorer

1.1 Inertial Explorer Overview

Inertial Explorer builds upon NovAtel's GNSS-only processor, GrafNav. Inertial Explorer shares a similar interface to GrafNav but also includes IMU processing capabilities. Both loosely coupled and tightly coupled are supported for both differential and Precise Point Positioning (PPP).

Inertial Explorer is well integrated with NovAtel SPAN products, however support is also available for processing third party IMU data. Inertial Explorer comes pre-configured with aerial, ground vehicle and marine processing profiles as well as a New Project Wizard that helps new customers get started quickly.

> This manual assumes the use of the *GrafNav/GrafNet 8.70 User Manual* which is available on our website at www.novatel.com/support/info/documents/572

1.2 Getting Started with Inertial Explorer

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

1.2.1 Installation

Verify that the installation was successful by ensuring that you have a *Waypoint Inertial Explorer 8.70* program group on your computer. If this program group is not there, refer to the *GrafNav/GrafNet 8.70 User Manual* for installation instructions.

1.2.2 How to start Inertial Explorer

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

1.2.3 Convert and Process GNSS Data

Refer to the *GrafNav/GrafNet 8.70 User Manual* to process GNSS data. The only exception is that the new project is created in Inertial Explorer, not GrafNav.

> For NovAtel SPAN users, all available GNSS, IMU, DMI, dual antenna heading and mount data will automatically be extracted by the GNSS converter.

1.2.4 Convert IMU Data

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

> NovAtel SPAN users do not have to follow these steps because this is done automatically when converting the raw GNSS data.

1.2.4.1 How to convert SPAN IMU data

1. Open the *Convert Raw GNSS to GPB* utility through *File* | *Convert* | *Raw GNSS to GPB*.
2. Use the *Get Folder* button to navigate to the directory containing the raw data.
3. Use the *Auto Add All* button to automatically add any detected raw GNSS data files to the *Convert Files* list.
4. Select *Convert*. All available GNSS, INS, DMI and dual antenna heading data will be converted.

1.2.4.2 How to convert third party IMU data

> Before using the *Waypoint IMU Data Conversion* tool, a generic IMU file must be formed as this is used as input. See *Inertial Explorer Data Formats* on page 43 for the format of this file.

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 conversion profile.
4. Click *Convert* to create the IMR file. See *Raw IMU Data Converter* on page 40 for more information.
5. Add the file to the project via *File* | *Add IMU File*.

1.2.5 Process SPAN IMU Data

The steps for processing SPAN IMU data are below.

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. Ensure that an appropriate processing profile for your application (aerial, ground vehicle, marine or pedestrian) and your SPAN IMU type has been automatically selected. If this needs to be changed, access the profile pull down menu.
3. Ensure the IMU to GNSS lever arm has been correctly loaded. If the IMU to GNSS lever arm was not saved during data collection, enter it.
4. Ensure the body to IMU rotation has been correctly loaded. If the body to IMU rotation was not saved during data collection, enter it.
5. Click *Process*.

The *Process* option on the LC and TC catalog features two additional options:

Process without pre-processing:

Using this option skips Inertial Explorer's pre-processing checks prior to processing. It may be necessary to use this option if you would like to ignore a critical pre-processing warning which disables processing.

Solve Lever Arm:

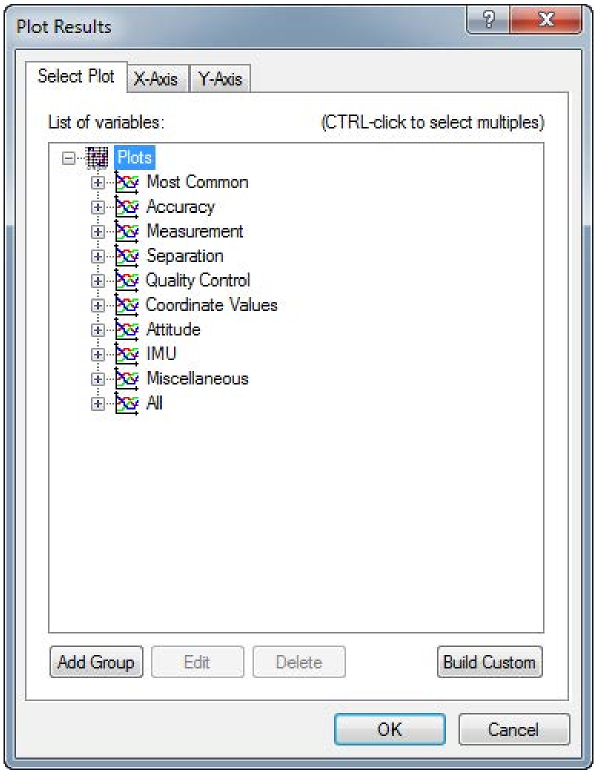
Using this option adds the X, Y and Z IMU to GNSS lever arm states to Inertial Explorer's Kalman filter. After processing and smoothing (if automated smoothing has been enabled within the *Solution* tab of *Settings* | *Preferences*), the best converged estimate of the lever arm is automatically reported after processing.

If processing *both* or *multi-pass* directions, both forward and reverse estimates are reported. Please note that several iterations may be required for the lever arm to converge and Inertial Explorer's ability to estimate lever arm values is dependent on the amount of data collected and vehicle dynamics.

If processing third party IMU data, an IMU error model must first be developed. It is also recommended to create a custom processing profile in order to automatically load all preferred processing settings.

1.2.6 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:



1.2.6.1 Attitude (Roll and Pitch)

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

1.2.6.2 Attitude (Azimuth/Heading)

This plot shows the heading/azimuth of the IMU and the GNSS course-over-ground (COG). They should be in reasonable agreement when the vehicle is moving forward.

If the GNSS COG and the IMU azimuth are biased by a large and constant amount (i.e. +/- 90 degrees or 180 degrees) it indicates the IMU sensor frame has not been rotated to the vehicle body frame (Y-forward, X-right and Z-up). If this is not intentional, a body to IMU rotation should be applied in the LC or TC processing dialog boxes. This will ensure Inertial Explorer's output roll, pitch and heading values are referenced to the vehicle frame.

1.2.6.3 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 to a reasonable level considering the quality of the IMU and the dynamics of the survey.

1.2.6.4 IMU-GNSS Position Misclosure

This plot shows the difference between the GNSS solution projected to the IMU center of navigation and the mechanized INS positions obtained from GNSS/INS processing. They should agree to a reasonable level, largely dependent on the quality of the GNSS trajectory (i.e. severity of the GNSS signal conditions).

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

> Consider adding your preferred Q/C plots to a group using the *Add Group* button. Your custom plot group will appear under the *Grouped Plots* list. When plotting a group, all plots within that group are simultaneously plotted.

1.2.6.5 Smooth Solution

By default, Inertial Explorer's backward smoother is automatically run after processing. This is needed to produce the best possible solution.

Automatic smoothing can be controlled through an option within the *Solution* tab of *Settings | Preferences*. If this has been disabled, it is strongly recommended to run the Smoother (*Process | Smooth Solutions*) prior to using the Export Wizard.

1.2.7 Export Final Coordinates

The steps for exporting final coordinates are below.

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.
4. Click *Next*. The Export Wizard will prompt you for all necessary information depending on the contents of your export profile.

> If the Export Wizard prompts you for a geoid, they can be downloaded from the NovAtel website at: www.novatel.com/support/waypoint-support/waypoint-geoids/.

1.3 File Menu

Refer to the *GrafNav/GrafNet 8.70 User Manual* 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

1.3.1.1 Project Wizard

The Project Wizard offers you a guided step-by-step way of creating a project. The *Project Wizard* steps are listed below.

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.

1.3.2 Add Master File(s)

Up to 32 base stations can be added to a single Inertial Explorer project. We recommend adding additional base station data only if each base station is in a distinctly different project area and is at some point the closest in the trajectory.

All data must be converted to GPB prior to adding as a base station. When adding a base station, take care to verify base station coordinates and datum (and epoch, if necessary) as this is critical to absolute position accuracy.

1.3.3 Add Remote File

Only one remote file can be added to an Inertial Explorer project. The file must be converted to GPB prior to adding it to the project. When selecting the remote GPB file, Inertial Explorer will automatically check for any associated IMU data (*.imr file), DMI data (*.dmr file), heading data (*.hmr file) and mount data (*.mmr) and prompt you whether you would like to add this data to the project as well.

When adding a remote GPB file, ensure the *Measured height* of the antenna is set to zero. Inertial Explorer uses the entered IMU to GNSS lever arm in order to transfer the GNSS position updates to the IMU center of navigation during processing. A vector can be entered from the IMU to any other sensor or point of interest on the vehicle during Export to transfer position data.

1.3.4 Add IMU File

Only one IMR file can be added to an Inertial Explorer project. This menu item is not often needed as IMR data will be automatically added to the project when adding the remote GPB file, provided it is in the same directory and has the same name as the remote GPB file.

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

1.3.5 Load

1.3.5.1 LC Solution (Loosely Coupled)

Loads the loosely coupled solution.

1.3.5.2 TC Solution (Tightly Coupled)

Loads the tightly coupled solution.

1.3.6 Convert

1.3.6.1 Raw GNSS to GPB

Raw GNSS data must be converted to GPB format for processing. See Convert Raw GNSS data to GPB in the GrafNav section for more information.

1.3.6.2 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 13 for more information.

This option does not need to be used by NovAtel SPAN customers. All NovAtel data (including all raw GNSS and IMU data) is automatically converted within the Raw GNSS Data Converter.

1.4 View Menu

Refer to the *GrafNav/GrafNet 8.70 User Manual* for a description of all the features available in this menu.

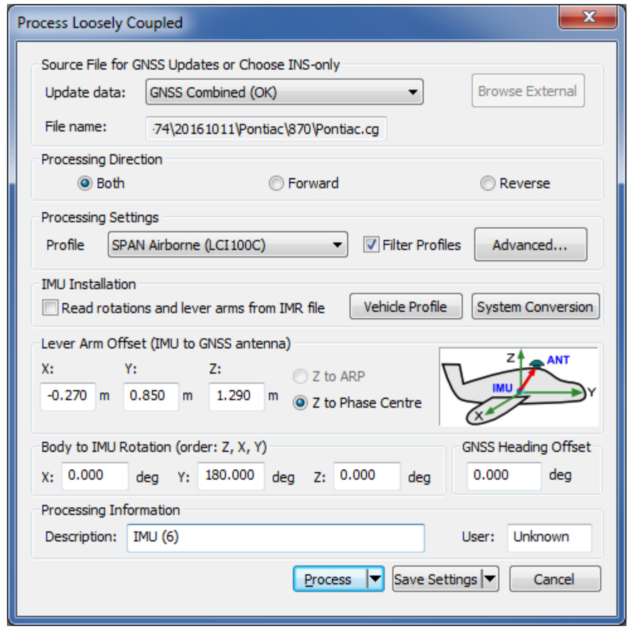
> In Inertial Explorer, view IMU message logs and trajectory files under *View* | *Forward Solution* and *View* | *Reverse Solution*. For information on file formats see *IMR File* on page 44, *DMR File* on page 46 and *HMR File* on page 48.

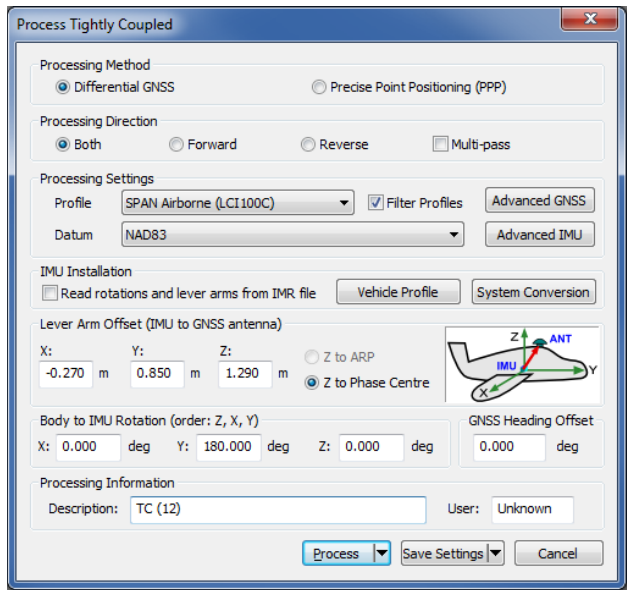
1.5 Process Menu

Refer to the *GrafNav/GrafNet 8.70 User Manual* 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.





Update Data

Use this option to select the GNSS file from which Inertial Explorer obtains updates. In most cases, the differential 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.

1.5.1.1 Process Settings

Profile

A processing profile is automatically loaded based on the detected processing environment (airborne, marine, ground vehicle, pedestrian) when converting the raw GNSS data to GPB format and the type of SPAN system used. If the automatically detected processing profile is incorrect it can be changed by accessing the pull down menu.

Filter Profiles

This option, enabled by default, will only show profiles associated with the SPAN IMU in use. If using third party IMU data, this option should be disabled in order to gain access to custom processing profiles.

Advanced...(LC Processing)

This button provides access to all IMU processing settings.

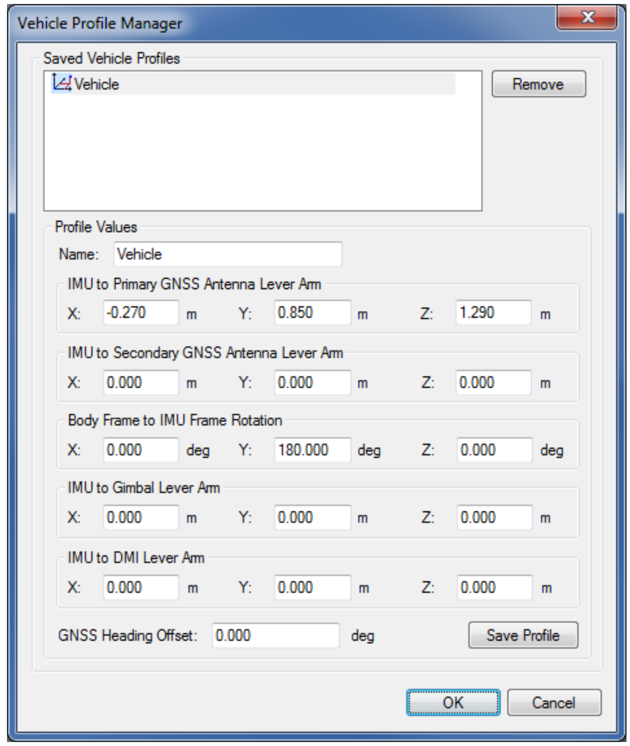
Advanced GNSS (TC Processing)

This button provides access to all GNSS processing settings. Refer to the *GrafNav/GrafNet 8.70 User Manual* for information.

Advanced IMU (TC Processing)

This button provides access to all IMU processing.

1.5.1.2 IMU Installation



Read rotations and lever arms from IMR file

If using a NovAtel SPAN system, the IMU to GNSS lever arm and body to IMU rotation may be set during data collection. If this has been set and the necessary logs (IMUTOANTOFFSETSB, VEHICLEBODYROTATIONB and SETIMUORIENTATIONB) are present in the raw data, this information is imported automatically to Inertial Explorer.

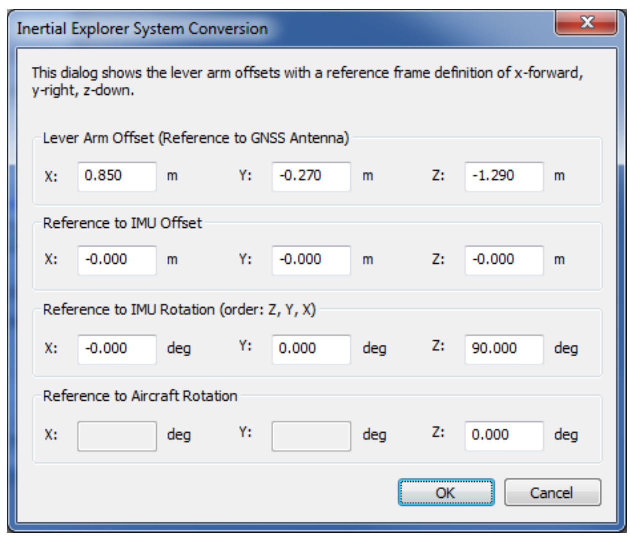
Vehicle Profile

This button accesses the Vehicle Profile Manager. It allows the primary lever arm, secondary lever arm, body to IMU rotation, gimbal lever arm, DMI lever arm, and GNSS heading offset to be saved to a vehicle profile. This facilitates quick and easy loading of important project parameters that are specific to each survey vehicle.

It is not necessary to save vehicle profiles if using a NovAtel SPAN system as this information can be retrieved directly from the raw data and imported automatically. Vehicle profiles are intended to assist non-SPAN customer workflow.

System Conversion

Inertial Explorer uses a vehicle frame of Y- forward, X-right and Z-up. This is different from other GNSS/INS processing software packages which use X-forward, Y-right and Z-down. If you are used to working in the latter system and are transitioning to Inertial Explorer, the System Conversion tool allows you to input your project parameters within the X-forward, Y-right and Z-down frame and converts the values to the conventions used by Inertial Explorer.



1.5.1.3 Lever Arm Offset (IMU to GNSS Antenna)

To perform GNSS updates accurately, enter the 3-D offset, in metres, from the IMU sensor array's navigation center to the GNSS antenna. This offset vector must be entered with respect to the body-frame of the vehicle, as *Figure 1: Body Frame Definition for Lever Arm Offset* on the next page shows.

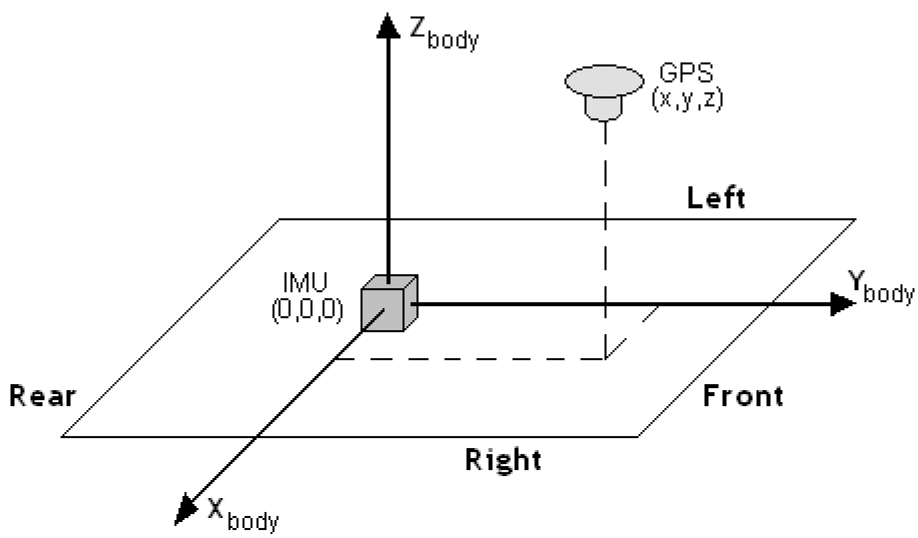
You must also specify whether the Z value applies to the antenna's reference point (ARP) or L1 phase center. To specify ARP, you must select an antenna model when you add the remote GPB file to the project. In this case, the antenna model's offset value is applied to the Z value to raise the Z value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

> Save lever arms for future access using the *Vehicle Profile* button.

Read from IMR file

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

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.

1.5.1.4 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.

1.5.1.5 GNSS Heading Offset

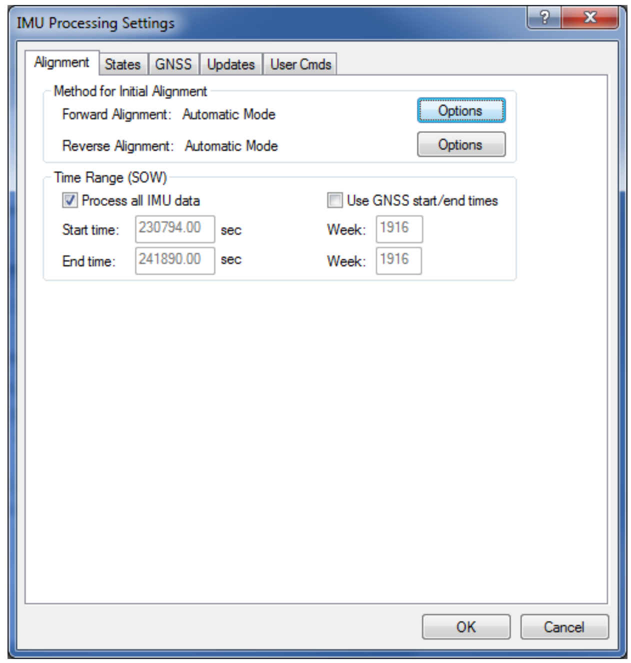
This value may also be referred to as the "Reference to Aircraft Rotation" and is meant for customers that have backwards-pointing LIDAR applications or other specialized applications where the IMU cannot be rotated to the vehicle frame through body to IMU rotations. This option applies a correction (as entered in degrees) to GNSS Course Over Ground (COG) values in order to allow kinematic alignments to succeed when the IMU is intentionally not aligned to the vehicle frame. This option has no effect if a static alignment is performed.

1.5.1.6 Advanced IMU

This button provides access to all IMU processing options.

1.5.1.7 Alignment

Method for Initial Alignment



In INS processing, small changes in velocity and orientation are integrated in order to derive position, velocity and attitude from a starting point. As such, it is a relative positioning method and the initial integration conditions must be known. Alignment is the process of solving these initial integration constants.

The initial position and velocity of the IMU are usually derived from Inertial Explorer's GNSS processor. Initial roll and pitch are derived from the accelerometer measurements and initial heading is derived from gyroscope measurements.

IMUs of tactical grade or higher are capable of static alignment. However MEMS IMUs, or any IMU with a gyro bias larger than the Earth rate (15 deg/hr at the equator), are not capable of deriving a reliable heading from gyro measurements alone. In these cases, the GPS Course-Over-Ground (COG) must be used to help Inertial Explorer determine the initial azimuth of the IMU.

Click the *Options* button to open the *Align Options* dialog. See *Align Options* on the next page for information about the settings available on this dialog.

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.

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.

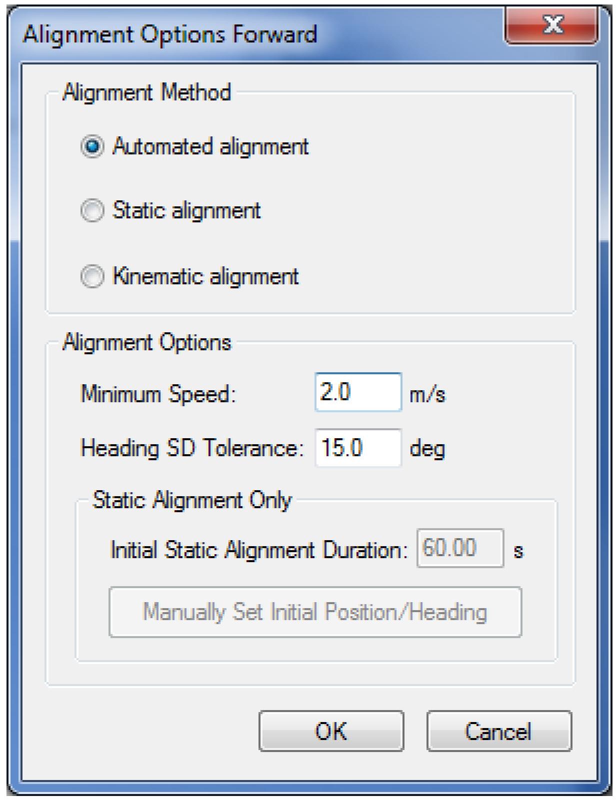
Start Time
Forward alignment will begin at the entered time (GPS SOW). Reverse processing will end at this time.

End Time
Reverse alignment will begin at the entered time (GPS SOW). Forward processing will end at this time.

Align Options

Alignment Method

Automated Alignment (Recommended)
Automated alignment scans the raw IMU data in order to determine whether a static alignment can be attempted. If no usable static alignment is detected, a kinematic alignment is applied when the vehicle reaches the minimum speed set within the auto/kinematic align tolerance. If static data is detected, a static coarse alignment is attempted for the duration of the detected session length.



Static alignment
Static 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. As such, only IMUs with gyro biases much less than the Earth rate (15 deg/hr at the equator) are capable of reliable static alignment.

Kinematic alignment
When no static data is detected, a kinematic alignment will be used. The GNSS Course-Over-Ground (COG) will be used as an initial approximation of the forward pointing IMU axis. As such, it is important when using a kinematic alignment that the IMU be mounted Y-forward, X-right and Z-up. If it is not possible to mount the IMU in this fashion, appropriate body to sensor rotations should be entered. If intentionally misaligning the sensor and vehicle frames, use the GNSS Heading Offset to correct the GNSS COG used in the alignment process. Kinematic alignment requires that the IMU be traveling relatively straight and level for at least 4 seconds.

Alignment Options

Minimum Speed
Specifies the minimum speed that the system must be traveling before kinematic alignment is attempted. This value should be lowered for low dynamics applications such as pedestrian surveying. Minimum speed is not considered for static alignments.

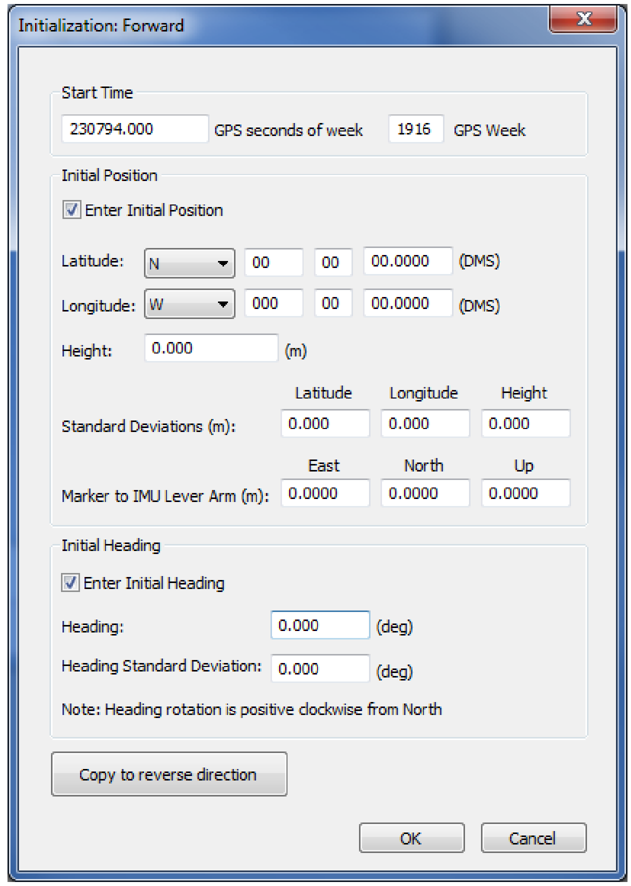
Heading SD Tolerance
Specifies the tolerance below which the heading standard deviation must fall before the alignment routine will move into navigation mode. Lower this value if the software is not achieving a good alignment. Raise this value if the software is not aligning at all. This value is tested for both static alignments and a kinematic alignments.

Initial Static Alignment Duration

Specifies the length of time the system must be stationary for static alignment. If you do not know this value you can check the GNSS velocity plot.

Manually Set Initial Position/Heading

This dialog allows the user to manually set the initial position and heading for a static alignment. It is particularly useful for IMU-only processing or denied GNSS environments. The position and heading values entered should be known to a good degree of accuracy.



The *Start Time* will automatically be loaded depending on the start/end time in the IMR file.

The *Initial Position* is to be entered in DMS format with ellipsoidal height. The *Standard Deviations* values are mandatory. Enter small values if the accuracy of the input position is well known. The *Marker to IMU Lever Arm* values are optional. They are intended to be used if the unit is starting near a known point. The lever arm must be entered in the local level (ENU) frame. A compass or magnetometer may be used, for example, to determine the East and North directions.

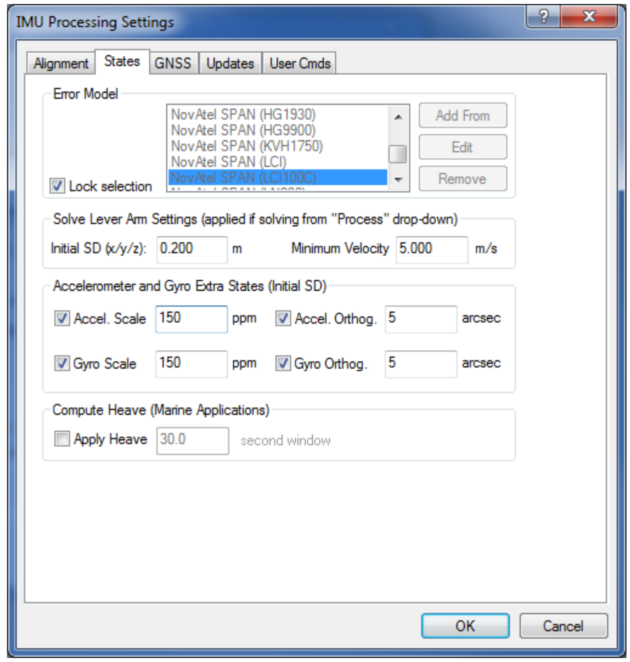
Initial Heading is to be entered in decimal degrees along with its standard deviation. If the heading is well known, enter a small value for standard deviation. The initial heading can be gathered using a compass or magnetometer, for example.

Positive heading rotation is clockwise from North

The *Copy to <opposite> direction* button may be used to copy the input parameters to the opposite direction for alignment (i.e. from forward to reverse or vice versa). This is useful if the beginning and end of the survey are on an identical position and identical orientation.

1.5.1.8 States

Error Model



Error models consist of initial standard deviation and spectral density values which are indicative of IMU sensor quality. They help control the extent to which Inertial Explorer weights GNSS and INS measurements during processing.

Inertial Explorer comes pre-configured with error models for all SPAN IMUs. These are automatically selected when loading an appropriate processing profile for the application (aerial, ground vehicle, marine and pedestrian) and SPAN system used. Inertial Explorer also comes pre-configured with a handful of error models developed for third party IMUs as well as generic error models that can be a useful starting point when attempting to develop your own new error model for a non-SPAN IMU. See *Edit Error Model Values* on the next page for information about the settings you can change in the Error Model.

Solve Lever Arm Settings

These values are applied if using the *Solve Lever Arm* feature under the *Process* pull-down menu on either the TC or LC processing dialog.

Inertial Explorer's ability to observe the IMU to GNSS lever arm is largely dependent on the length of data collection, quality of GNSS data and vehicle dynamics. This feature is not meant as a substitute for measuring the lever arm but rather for checking or troubleshooting purposes.

Initial SD
This value reflects the uncertainty in the lever arm measurement.

Minimum Velocity
The lever arm state will not be updated unless this minimum velocity has been reached. A minimum value of at least 1 m/s is suggested (but not required) to help avoid the possibility of the IMU to GNSS lever arm state diverging under very low dynamics.

Accelerometer and Gyro Extra States

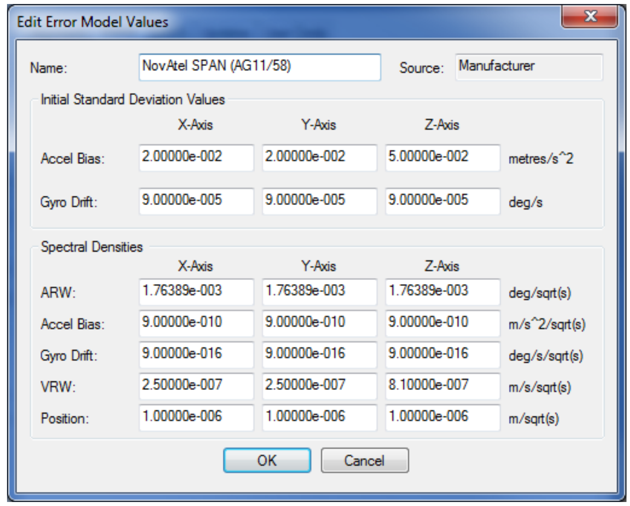
These options add scale and/or non-orthogonality states to the Kalman filter for the accelerometer and gyroscope measurements. They are often needed by MEMS sensors to account for errors in the manufacturing process.

Compute Heave (Marine Applications)

For marine users who wish to apply heave compensation to the computed ellipsoidal height, use this option to engage Inertial Explorer’s low-pass filter. The algorithm requires that a window size reflecting the period of the wave motion be entered. The smaller the window size, the more responsive the filter will be to the wave motion. If using this option, the computed ellipsoidal height and the heave compensated height can be viewed from the *Height Profile* plot. The computed heave value can also be viewed from the *Heave* plot.

After processing with *Apply Heave* enabled, you can access heave compensated ellipsoidal and orthometric heights from the Export Wizard. The *Height Profile* plot will display the processed ellipsoidal height together with the heave-compensated ellipsoidal height in order to show the effect of the heave window used. The longer the window used the smoother the marine heave compensated height will be (i.e., less responsive to changes).

Edit Error Model Values



Error model editing is only necessary when developing your own error model for a new IMU. Note that when working with MEMS sensors, it may also be necessary to enable accelerometer and gyro extra states in order to achieve a reasonable level of performance in addition to error model tuning.

Initial Standard Deviation Values

The following mathematical quantities are available:

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:

ARW (Angular Random Walk)
Angular Random Walk, 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 ARW, 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.

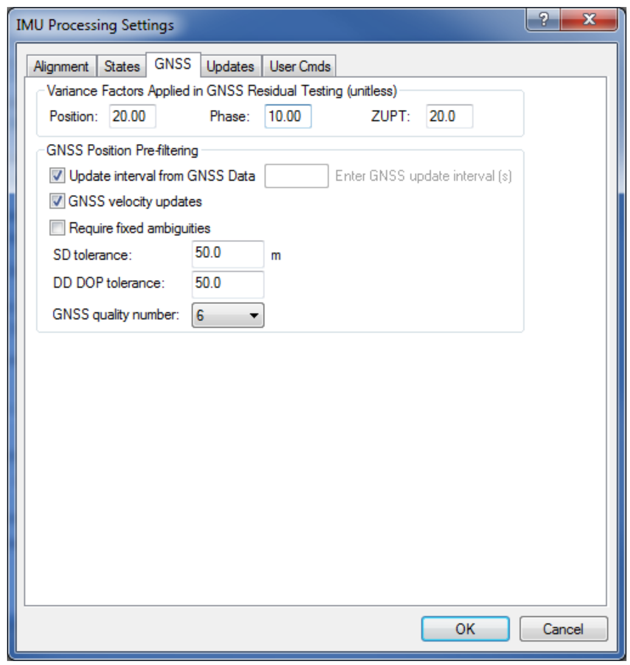
Gyro Drift
Gyroscope drift densities similarly act as additives to the covariances computed for the gyroscope drift states. In the case of inexpensive units, larger values here may be necessary.

VRW (Velocity Random Walk)
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.

1.5.1.9 GNSS

Variance Factors Applied in GNSS Residual Testing



Inertial Explorer performs residual testing using a standard least squares approach on every type of updated applied within our Kalman filter. Phase updates are only applied when there are a minimum of two satellites available in TC processing.

Updates are accepted only if the computed residual is within the set tolerance. The larger the variance factor tolerance, the less likely an update is to be rejected by residual testing. For this reason, large values are typically applied in aerial processing profiles in order to reduce the chances of false rejection and lower thresholds are applied in ground vehicle profiles in order to lessen the likelihood that a biased update will be accepted. It is safe to use large values in clean GNSS environments where the quality of GNSS data is good, however lower values (1-3) are recommended when surveying in challenging GNSS environments.

GNSS Position Pre-filtering

Update interval from GNSS Data
This option is only available in tightly coupled processing. By default, Inertial Explorer will process using all available GNSS data. This results in position updates being available at the nominal GNSS logging rate. Use this option to limit the GNSS processing interval in tightly coupled processing. This can be useful if logging GNSS data at high rates (i.e. >1 Hz).

GNSS velocity updates
GNSS velocity updates are important, especially when a kinematic alignment is performed. As such, this option is normally engaged. However, for any special applications where GNSS velocity updates are to be rejected, they can be disabled here.

Require fixed ambiguities
If using a high precision IMU and when surveying in urban conditions with some challenging GNSS data, this option may be useful in achieving the best possible results. This option is not recommended for most systems (MEMS or tactical grade IMUs) as any GNSS updates (even one derived from float ambiguities) are generally beneficial in observing IMU sensor errors.

SD tolerance
If the position update returned by the GNSS processor is larger than this value, it will not be passed to the IMU filter. By default a large threshold is used as Inertial Explorer relies on variance factor testing to determine whether a GNSS position update should be applied or not.

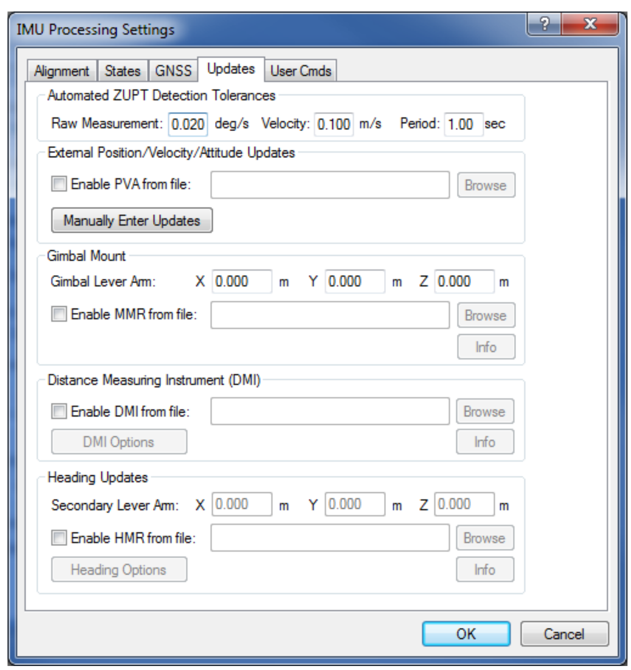
DD DOP tolerance
Double Differenced DOP (DD DOP) is roughly equivalent to PDOP squared. GNSS position updates with large DOP values can be unreliable, however Inertial Explorer by default uses a very loose pre-filtering threshold and instead relies on GNSS variance factor testing to determine whether or not a GNSS position update should be applied. If it is desired to use a lower value, enter it here.

GNSS quality number
The GNSS processing engine assigns a quality number to each processed epoch between the values of 1 to 6, 1 being the best. By default Inertial Explorer does not use the GNSS quality number in pre-filtering as instead it relies on GNSS variance factor testing in determining whether a GNSS position update should be accepted. If you wish to enable a lower prefiltering tolerance, enter it here.

1.5.1.10 Updates

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.

External Position/Velocity/Attitude Updates

A binary PVA file can be input to Inertial Explorer if external position, velocity and/or attitude updates are available. See *PVA File* on page 48 for the format of this file. Input of this binary file is the recommended approach if large numbers of external updates are available. If using a PVA file, the following user commands are supported:

- PVA_ENABLE = “NEVER”/”ALWAYS”/”SPOSSD” [tol (m)] (default is “ALWAYS”)
    - NEVER = never use update ([tol] = 0)
    - ALWAYS = always use update ([tol] = 0)
    - SPOSSD = use after solution standard deviation exceeds [tol]
- PVA_LEVER = “ON”/”OFF” [x] [y] [z] [def=OFF]
    - Shift position of update to IMU using body-frame lever arm values. Sensor to IMU (m)
    - This command will override any offsets in the binary PVA file
- PVA_MEAS_SF = sfpos sfvel sfatt [def=1 1 1]
    - Scale input standard deviations by these scale factors
- PVA_SIGMA_TOL = ON/OFF pos vel att [def=OFF]
    - Unitless a-posteriori standard deviation ratio for PVA for blunder detection
- PVA_UPDATE = “REC/POS|VEL|ATT” [def=REC]
    - REC = use record type from record
    - POS|VEL|ATT = Space separated mask for update type in file

User commands can be input through the *User Cmds* tab of the *Advanced IMU* options.

Relative Updates

In version 8.70, Inertial Explorer can accept relative position updates that have been measured by an external sensor, such as a camera-array and/or LIDAR sensor. For the photogrammetric case, knowledge of scale (or depth) is important, which becomes difficult with monocular vision systems. Therefore, stereo vision systems with sufficient base (camera separation) is suggested. In the case of LIDAR, a sufficient number of surfaces that are perpendicular to the direction of travel need to be present, or preferably estimated accuracy values will compensate for geometry variations. Regardless of the input source, the relative position inputs will have varying accuracy, depending on the number and geometrical distribution of matched points. Thus, computing representative, variable SD values is suggested.

Relative updates are measured vector components between two timed events, which are determined by an external system that is time-synchronized with the navigation system. Basically, the sensor measures the relative motion and orientation between the two epochs, which are generally 0.2 s to 2 s apart. The best data-rate for the formation of the updates and the best rate for input into Inertial Explorer may differ, where Inertial Explorer may benefit from a lower rate. The problem with very short intervals is that the noise (measurement error) can be a significant portion of the vector length. Therefore, an external pre-processor may need to accumulate high-rate measurements to a lower rate like 1 Hz, which is suitable for Inertial Explorer.

Coordinate Systems

Currently, two coordinate systems are supported for input:

- **Local Level**
  The input y-axis is aligned to true-north, x-axis to east and z-axis to up. This implies that the vectors have been aligned to a global coordinate system. In the absence of external updates or control points, a solution can drift, meaning the absolute orientation (i.e. 6-parameter transformation between the sensor frames) may vary. Furthermore, if a local rectangular system (LRS), which also called Local Cartesian in Inertial Explorer, is used and the mission covers a larger area (greater than a few kilometres), then a meridian convergence[1] compensation must be applied. In this mode, the relative attitude values are not used nor needed.
- **Body System**
  By keeping the measurements in the sensor frame of the measuring system, the task of relating the relative vector to the computational system, which is ECEF, is performed by Inertial Explorer's Kalman Filter. For datasets that are non-continuous (or have gaps), not having to reference the relative vector in Local Level can be a benefit. In Body processing, Inertial Explorer will account for it by using its internal, filtered, pre-smoothed estimates of the attitude, and their associated errors. Therefore, for continuous datasets, the translation from body to Local Level may best be performed external to Inertial Explorer using a smoothed trajectory. Finally, the body system of the IMU is normally referenced as y-forward, x-right and z-up, but the imaging/LIDAR sensor system may be different, and their axes may not be aligned. Hence, the relative alignment/boresight and offset of these systems must be accounted for before entry into Inertial Explorer.

Both these input methods ultimately utilize the update in Inertial Explorer's computational system, which is ECEF/e-frame. So the main difference is how the body-to-Local Level conversion is applied. The Local Level-to-ECEF depends only on position and is not very sensitive. However, body-to-Local Level is directly affected by the attitude accuracy. So, if the pre-PVA smoothed Inertial Explorer trajectory is best, use Local Level. If the filtered Inertial Explorer trajectory using PVA updates is best, use body.

Data Input

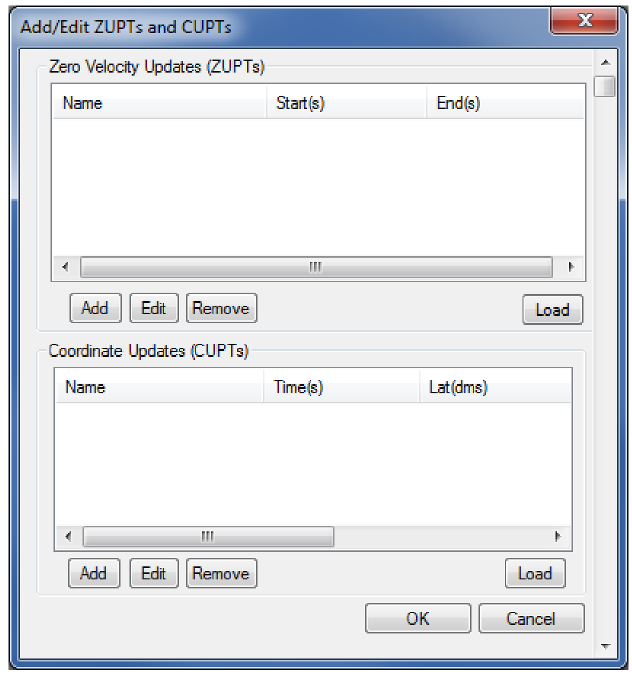
- The relative updates need to be converted to a binary PVA file before processing in Inertial Explorer. See *PVA File* on page 48 for the format of this file. The following variables are needed:
- Start TOW, week – Begin GPS time of the relative update (epoch i-1)
- End TOW, week – End time (epoch i), called update time in the format below
- X, Y, Z – relative vector components (m) (Body or Local Level)
- SDx, SDy, SDz – estimate accuracy of vector components (m)
- Covariance/CC – off-diagonal covariance values or correlation coefficients (CC). CC values are stored in PVA file. If the values are not available, leave them as zeros.
- tx,ty,tz – relative orientation between two epochs in the frame of epoch i/end defined as roll, pitch, yaw. These are used only for body-frame processing in reverse because in the forward direction the XYZ relative vector is already in the correct frame.

Important points to consider are:

[1] Meridian convergence is the angular orientation difference between grid north and true north

- Knowing or estimating the accuracy of the vector components is extremely valuable, and is preferred over the just using constant values.
- The relative orientation values are not the absolute angles and can be complicated to produce. The values stored in the PVA file are very specifically defined in the roll-pitch-yaw frame aligned but aligned to the end-time, as this is when the update is performed (for forward processing). They are ignored for Local Level and body-forward processing. Therefore, if in body mode the forward solution is satisfactory but the reverse has higher than expected errors, then the relative tx,ty,tz angles may not be correct.
- The accuracy of the relative orientation values are not needed nor used.

Manually Enter Updates



Inertial Explorer also accepts input of individual ZUPTs and CUPTs through the *Manually Enter Updates* button. This feature is more convenient for customers that want to manually enter a small number of such updates as opposed to having to format your own binary file. This feature also supports loading of ZUPTs and CUPTs from an ASCII file, however this feature is limited to supporting a maximum of 1000 updates. For this reason, using the binary PVA file is encouraged when large numbers of external updates exist.

Zero Velocity Updates (ZUPTs)

Inertial Explorer automatically detects 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. Nonetheless, individual ZUPTs can be added here or loaded from an ASCII file.

Coordinate Updates (CUPTs)

External coordinate updates can be very beneficial to GNSS/INS post-processing in areas of denied GNSS signal reception if they can be properly time tagged. This dialog can be used to add individual time coordinate updates or load from an ASCII file.

Gimbal Mount

If using a gimbal mount, the IMU to gimbal center lever arm can be entered here. This should be entered in the vehicle frame (Y-forward, X-right, Z-up) with the origin at the IMU center of navigation. This causes Inertial Explorer to shift it's output from the IMU to the Gimbal Center.

MMR files are automatically produced by the SPAN data converter and contain the rotations of the stabilized gimbal platform. This is required in order to properly compensate for the changing IMU to GNSS lever arm during operation of the gimbal unit.

Distance Measuring Instrument (DMI)

If logging DMI data, the NovAtel SPAN decoder will automatically write a DMR file which contains time stamped DMI measurements. If a DMR file is detected during project creation, it is automatically loaded into the project and thus does not have to be explicitly set here.

DMI Options

> A typical DMI will either output a tick count or a measured speed. If tick counts are recorded, Inertial Explorer converts them as a velocity update. If speed was recorded, then the software applies them as such.

Detect ZUPTs from DMI sensor
This option is off by default as Inertial Explorer already has two layers of ZUPT detection; analysis of the raw IMU measurements and using available GNSS data. This option therefore is generally not needed and if the DMI used does not function well at low velocity, it can actually be harmful.

If however a high resolution DMI is used which works well at low velocity and if ZUPTs will be observed during periods of extended GNSS outages, this option can be very beneficial in helping to observe ZUPTs.

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 wheel circumference is extracted from raw SPAN data if the WHEELSIZEB log is present. The default value is 1.96 metres if no value is detected from the raw GNSS data or set during conversion. Inertial Explorer computes a DMI scale factor to account for varying wheel sizes during data collection, however the best estimate possible of the wheel circumference should be input.

Heading Updates

If using the NovAtel dual antenna ALIGN system and requesting HEADINGB logs, an HMR file is automatically produced during data conversion which can be input to Inertial Explorer here. The secondary lever arm must also be set if using it in *Dual GPS Antenna System* mode.

Heading updates are most useful in assisting auto/kinematic alignment in low dynamic applications such as marine surveys. When a kinematic alignment is used and heading updates are available, Inertial Explorer will extract the initial heading of the vehicle from the HMR file.

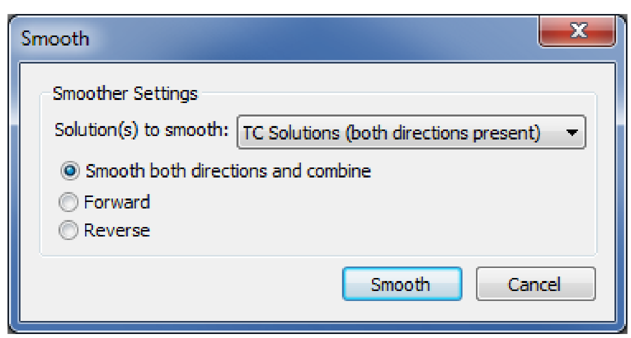
The HMR data format is described in *HMR File* on page 48.

1.5.2 Combine Solutions

Refer to the *GrafNav/GrafNet 8.70 User Manual* for information regarding these options. Only points relevant exclusively to Inertial Explorer are made here.

1.5.2.1 Smooth Solutions

Inertial Explorer is capable of combining processing directions and/or performing back smoothing on an inertial trajectory.



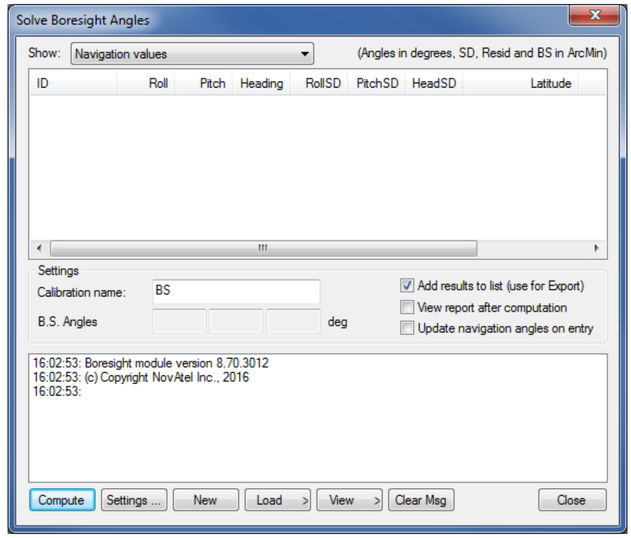
Smoothing greatly improves position quality over GNSS data gaps. The benefits of smoothing are not limited to improving position quality however; velocity and attitude are also back-smoothed. Even if no GNSS data gaps are observed, smoothing will always generate the highest quality trajectory possible.

Smoother Settings

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. Smoothing both directions will also generate the combined smoothed trajectory.

1.5.3 Solve Boresighting Angles

1.5.3.1 Show



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

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.

1.5.3.2 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.

1.5.3.3 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.

Settings...
This button gives access to the *Boresight Settings* window, which is useful for configuring many parameters used in the boresight calibration. See *Axes/System Definition* on the next page for information about the setting on this window.

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.

1.5.3.4 Boresight Settings

Axes/System Definition

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.

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.

1.6 Settings Menu

Refer to the *GrafNav/GrafNet 8.70 User Manual* for information regarding all of the features available from this menu.

1.7 Output Menu

Refer to the *GrafNav/GrafNet 8.70 User Manual* 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 8.70 User Manual* 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.

The following table contains descriptions of the IMU plots available only through Inertial Explorer.

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 vehicle 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 a crab angle is visible in this plot if the GNSS COG bears a constant offset from INS heading. The *IMU Heading COG Difference* 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.

Plot	Description
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.
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.
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.
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.

Plot	Description
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.

1.7.2 Export Wizard

Only the *Export Wizard* window exclusive to Inertial Explorer is discussed here. Refer to the *GrafNav/GrafNet 8.70 User Manual* for additional information concerning this feature.

1.7.2.1 IMU Epoch Settings

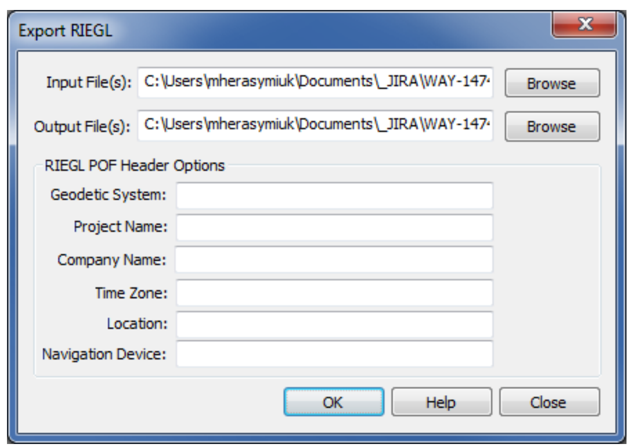
Limit Exported Time Range
The time range to export can be changed here. Multiple time ranges may be entered.

Epoch Interval Options
The *Binary Trajectory Interval* displays the interval at which the IMU data was processed. It is the smallest interval that can be output without interpolation. The *Time Interval* option will export data at the input interval. It may be set to a value as small as 0.001 seconds (1000 Hz). Any value smaller than, or not a multiple of, the *Binary Trajectory Interval* will require interpolation of data. The *Distance Interval* option will output data whenever the input distance threshold is met/exceeded.

Transfer IMU Coordinates
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.

1.7.3 Export to RIEGL POF/POQ



Selecting this option launches the *Export RIEGL* dialog which supports the conversion of Inertial Explorer's native binary output to RIEGL POF (Position & Orientation file) format. The POF file format is supported by third party software manufacturers and as such is a convenient way to integrate Inertial Explorer into a larger workflow.

1.7.4 Export to SBET

Selecting this option launches the *Export SBET* dialog which supports the conversion of Inertial Explorer's native binary output to Applanix SBET binary format. The Applanix SBET file format is supported by many third party software packages and thus this utility is a convenient way to integrate Inertial Explorer into a larger workflow.

1.7.5 Export to Waypoint Legacy Format

This option converts your Waypoint trajectory files to the binary format defined by version 8.60 of the software. It is useful if you have a workflow catered to the 8.60 SBTC/SBIC data format.

> This tool converts the trajectory currently loaded in the map window. Load the desired trajectory before converting to Waypoint legacy format.

1.8 Tools Menu

Refer to the *GrafNav/GrafNet 8.70 User Manual* for information regarding all of the options available via this menu.

1.9 Interactive Windows

Refer to the *GrafNav/GrafNet 8.70 User Manual* for information regarding the *Map Window* and the features available within it.

1.10 Processing Window

Table 1: IMU Plots on page 36 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 8.70 User Manual* 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.

1.12 About Inertial Explorer

This window displays information about the software version, build dates and copyright information.

Chapter 2 Conversion Utilities

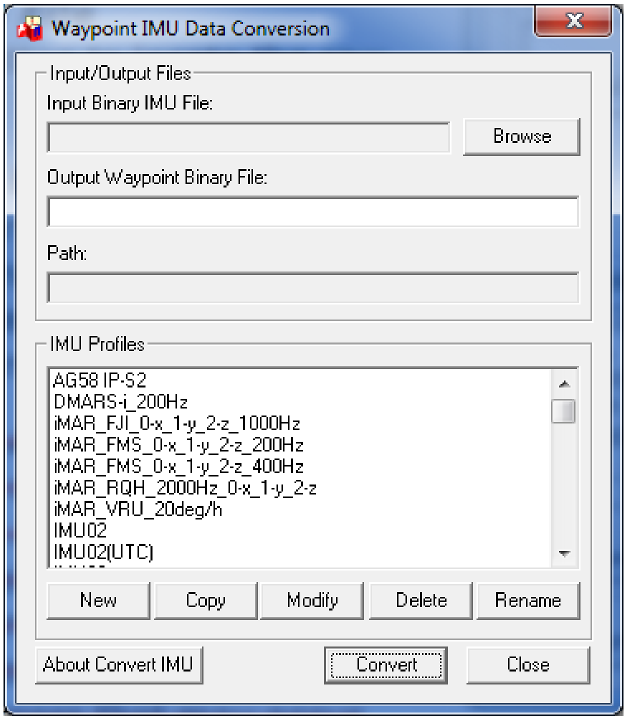
2.1 Raw IMU Data Converter

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

2.1.1.1 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.

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

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.

Copy
Copies an existing profile to a new name. Useful if you want to modify an existing profile without overwriting it.

Modify
Allows changes to be made to an existing profile.

Delete
Deletes an existing profile.

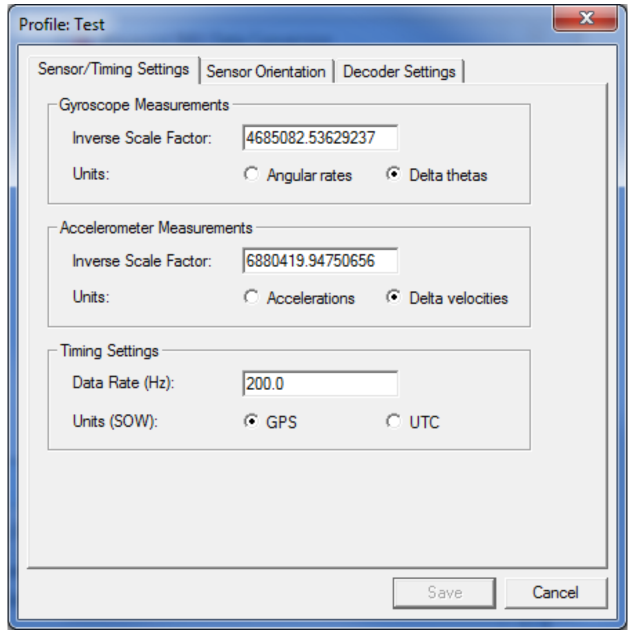
Rename
Renames an existing profile.

2.1.1.2 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 *Creating / Modifying a Conversion Profile* below 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

2.1.2.1 Sensor/Timing Settings



Gyroscope 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*.

Accelerometer 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 *Delta velocities*, and the other being *Accelerations*.

Timing Settings

Enter the data collection rate of the IMU sensor and specify the time system (GPS time or UTC time) of the IMU measurements.

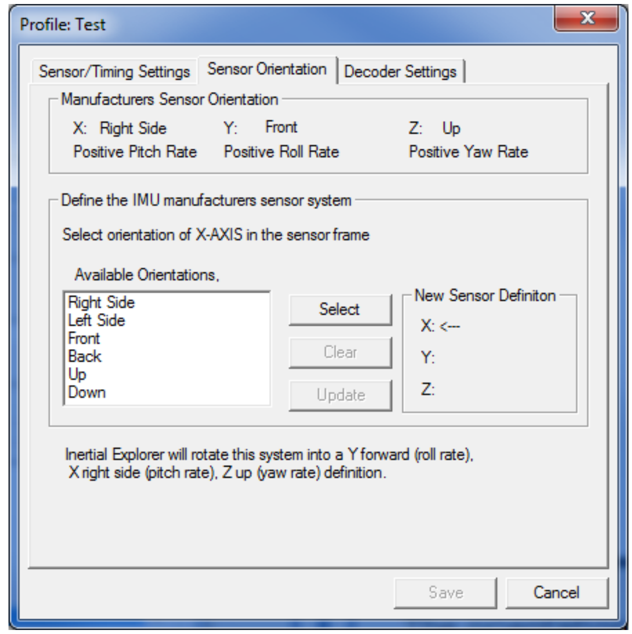
2.1.2.2 Sensor Orientation Settings

Define the orientation of the IMU here using the steps below.

The orientation will always be right-handed.

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, the z-axis 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.

2.1.2.3 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.

Chapter 3 Data and File Formats

3.1 Inertial Explorer 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 *Waypoint IMU Data Conversion* on page 40.

The following table presents the binary structure in which the conversion utility expects the raw IMU data to be logged.

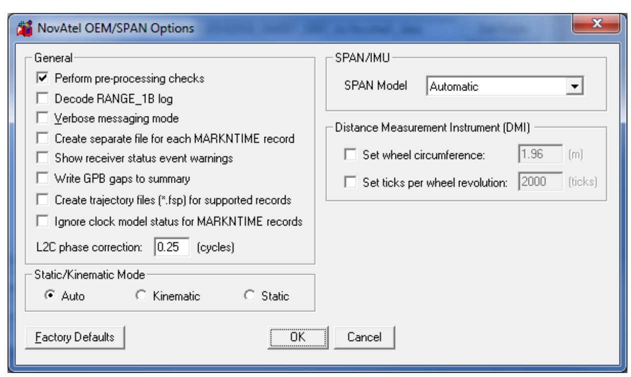
> The variable types in the table below are taken from the C standard library header "cstdint"/"stdint.h".

Table 2: Binary Structure of Raw Data

Word	Size (bytes)	Type	Description
GpsTime	8	double	time of the current IMU rate measurements in GPS seconds of the week
GyroX	4	int32_t	scaled X-body axis gyro measurement as an angular increment or angular rate
GyroY	4	int32_t	scaled Y-body axis gyro measurement as an angular increment or angular rate
GyroZ	4	int32_t	scaled Z-body axis gyro measurement as an angular increment or angular rate
AccelX	4	int32_t	scaled X-body axis accelerometer measurement as a velocity increment or acceleration
AccelY	4	int32_t	scaled Y-body axis accelerometer measurement as a velocity increment or acceleration
AccelZ	4	int32_t	scaled Z-body axis accelerometer measurement as a velocity increment or acceleration

3.1.1 NovAtel's SPAN Technology

With the use of NovAtel's SPAN technology, all GNSS and IMU data decoding can be done in one step. 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 IMR 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 OEM/SPAN.* Then, click either the *Global Options* or *Options* button to gain access to the IMU decoding settings.

3.2 Inertial Explorer 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 *Raw IMU Data Converter* on page 40 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:

Table 3: IMR Header Struct Definition

Word	Size	Type	Description
szHeader	8	char[8	"$IMURAW\0" – NULL terminated ASCII string
bIsIntelOrMotorola	1	int8_t	0 = Intel (Little Endian), default 1 = Motorola (Big Endian)
dVersionNumber	8	double	Inertial Explorer program version number (e.g. 8.70)
bDeltaTheta	4	int32_t	0 = Data to follow will be read as scaled angular rates 1 = (default), data to follow will be read as delta thetas, meaning angular increments (i.e. scale and divide by *dDataRateHz* to get degrees/second)
bDeltaVelocity	4	int32_t	0 = Data to follow will be read as scaled accelerations 1 = (default), data to follow will be read as delta velocities, meaning velocity increments (i.e. scale and divide by *dDataRateHz* to get m/s2)
dDataRateHz	8	double	The data rate of the IMU in Hz. e.g. 0.01 second data rate is 100 Hz
dGyroScaleFactor	8	double	If *bDeltaTheta* == 0 multiply the gyro measurements by this to get degrees/second If *bDeltaTheta* == 1 multiply the gyro measurements by this to get degrees, then use *dDataRateHz* to get degrees/second

Word	Size	Type	Description
dAccelScaleFactor	8	double	If *bDeltaVelocity* == 0 multiply the accel measurements by this to get m/s2 If *bDeltaVelocity* == 1 multiply the accel measurements by this to get m/s, then use *dDataRateHz* to get m/s2
iUtcOrGpsTime	4	int32_t	Defines the time tags as GPS or UTC seconds of the week 0 = Unknown, will default to GPS 1 = Time tags are UTC seconds of week 2 = Time tags are GPS seconds of week
iRcvTimeOrCorrTime	4	int32_t	Defines whether the time tags are on the nominal top of the second or are corrected for receiver time bias 0 = Unknown, will default to corrected time 1 = Time tags are top of the second 2 = Time tags are corrected for receiver clock bias
dTimeTagBias	8	double	If you have a known bias between your GPS and IMU time tags enter it here
szImuName	32	char[32]	Name of the IMU being used
reserved1	4	uint8_t[4]	Reserved for future use
szProgramName	32	char[32]	Name of calling program
tCreate	12	time_type	Creation time of file
bLeverArmValid	1	bool	True if lever arms from IMU to primary GNSS antenna are stored in this header
lXoffset	4	int32_t	X value of the lever arm, in millimeters
lYoffset	4	int32_t	Y value of the lever arm, in millimeters
lZoffset	4	int32_t	Z value of the lever arm, in millimeters
Reserved[354]	354	int8_t[354]	Reserved for future use

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:

Table 4: IMR Record Struct Definition

Word	Size	Type	Description
Time	8	double	Time of the current measurement
gx	4	int32_t	Scaled gyro measurement about the IMU X-axis
gy	4	int32_t	Scaled gyro measurement about the IMU Y-axis

Word	Size	Type	Description
gz	4	int32_t	Scaled gyro measurement about the IMU Z-axis
ax	4	int32_t	Scaled accel measurement about the IMU X-axis
ay	4	int32_t	Scaled accel measurement about the IMU Y-axis
az	4	int32_t	Scaled accel measurement about the IMU Z-axis

The angular increments (or angular rates) are signed integers. The scale factor to obtain a double precision word must be supplied by the *dGyroScaleFactor* variable in the IMR header. Similarly, the accelerations (or velocity increments) are signed integers and must be scaled by the *dAccelScaleFactor* variable in the IMR 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.

Table 5: DMR Header Struct Definition

Word	Size	Type	Description
szHdr	8	char[8]	"$DMIRAW\0" – NULL terminated ASCII string
sHdrSize	2	int16_t	Size of the header in bytes, must be set to 512
sRecSize	2	int16_t	Size of each record (refer to *dmi_lrec_type* or *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
sValueType	2	int16_t	0 = logging data using 4 byte integer values 1 = logging data using double precision values
sMeasType	2	int16_t	1 = logging a distance measurement 2 = logging a speed measurement
sDim	2	int16_t	Number of DMI sensors. Max is 3, but only 1 can be used in Inertial Explorer
sRes	2	int16_t	Measurement resolution of the DMI 1 = low resolution, measurements on full wheel revolutions 2 = high resolution, measurements on partial wheel revolutions or fixed time intervals

Word	Size	Type	Description
sDistanceType	2	int16_t	Must be set if sMeasType == 1 1 = logging accumulated tick count 2 = logging distance in meters 3 = logging accumulated distance in meters
sVelocityType	2	int16_t	Must be set if *sMeasType* == 2 1 = logging velocity in metres/second 2 = logging velocity in ticks/second
dScale	8	double	DMI scale factor in metres/count or metres/second/count. Must be set if *SValueType* == 0 1.0 if logging accumulated tick count or ticks/second
szAxisName	48	char[3][16]	Name of axes of each DMI. Optional
dWheelSize	8	double	Circumference of the wheel sensor in metres. Must be set if logging accumulated tick count
lTicksPerRevolution	4	int32_t	Number of tick counts per revolution of the wheel sensor. Must be set if logging accumulated tick count or ticks/second
cExtra2	420	int8_t[420]	Reserved for future use

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

Table 6: DMR Long Record Struct Definition

Word	Size	Type	Definition
sSync	2	int16_t	Sync byte, 0xFFEE
sWeek	2	int16_t	GPS week number, set to -1 if unknown
dTime	8	double	GPS seconds into week
lValue	12	uint32_t[3]	Values for each DMI sensor as integers. Only use lValue[0] for Inertial Explorer

Table 7: DMR Double Record Struct Definition

Word	Size	Type	Description
sSync	2	int16_t	Sync byte, 0xFFEE
sWeek	2	int16_t	GPS week number, set to -1 if unknown
dTime	8	double	GPS seconds into week
dValue	24	double[3]	Values for each DMI sensor as double precision. Only use dValue[0] for Inertial Explorer

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:

Table 8: HMR Header Struct Definition

Word	Size	Type	Definition
szTitleStr	12	char[12]	"$IMUHEADING\0" – NULL terminated ASCII string
ucType	1	uint8_t	1 = values from external source 2 = values form dual antenna source
dBoreSightRotationZ	8	double	Heading boresight between the forward direction of travel and the vector between antennas. Positive rotation is clockwise about Z
dBoreSightRotationZStdDev	8	double	Accuracy of the boresight, 0.0 if not known
Extra	227	int8_t[227]	Reserved for future use

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

Table 9: HMR Record Struct Definition

Word	Size	Type	Definition
dGpsTime	8	double	GPS seconds of week
sGpsWeek	2	int16_t	GPS week
dHeading	8	double	The heading update value in decimal degrees, positive rotation is clockwise from North
fHeadingStdDev	4	float	Standard deviation of the update in decimal degrees., 0.0 if not known
fBaselineLength	4	float	Distance between antennas in meters. Only use if *ucType* == 2
fPitch	4	float	Pitch between the two antennas in decimal degrees. Only use if *ucType* == 2
fPitchStdDev	4	float	Standard deviation of the pitch in decimal degrees, 0.0 if not known

3.2.4 PVA File

Both PVA input and output data share the same binary file format. The PVA format has the following three records defined:

- **pva_hdr_type**
  Contains header information including coordinate frames/systems, version information, datum and position/orientation offsets.

- **pva_inprec_type**
  Contains the input data including precise time (TOW + week number), position & velocity in ECEF or Geographic of IMU, GNSS antenna or external sensor (where offsets are required). In addition, attitude can be specified in RPH. WPK is defined but not currently implemented. Using the *uUpdate* flag, each record may contain any combination of position, velocity and attitude. In addition to PVA values, standard deviation (SD) values are also specified while cross-correlation terms are also handled if they are available.
- **pva_outrec_type**
  Contains the processed results and is only present in the output from Inertial Explorer. It contains pre- and post-filter residuals, solution SDs and measurement SDs. Position and velocity output is currently always ECEF, which is how the Kalman filter handles them, while attitude is RPY.

An input [.PVA] file will have the following records:

- [pva_hdr_type]
- [pva_inprec_type]
- [pva_inprec_type]
- :

An output [.DTPVA] file (where D=(f)orward or (r)everse, and T=(t)ightly coupled, (l)oosely coupled), while PPP-TC is currently not implemented or defined). Each output file contains the following records:

- [pva_hdr_type]
- [pva_inprec_type]
- [pva_outrec_type]
- [pva_inprec_type]
- [pva_outrec_type]
- :

For each input, there will be one input + one output record. Note that an output file cannot be used for input, and *pva_outrec_type* structure is not described here.

3.2.4.1 PVA Binary Records

This section describes the binary structures used by PVA input.

Table 10: Binary Header Structure (pva_hdr_type)

Variable Name	Type	Description
szHdr[12]	char	Must be "$PVABIN\r\n\0"
usFileVersion	uint16_t	Currently 1
usHdrSize	uint16_t	Size of this header (bytes), normally 2048 bytes
usFileType	uint16_t	File type, input, output, see *pva_fileio_t*, INPUT=1, OUTPUT=10

Variable Name	Type	Description
usVersion[3]	uint16_t	Version numbers of creating program [major, minor, build]
szProgramName[32]	char	Name of program that created file
ulNumRec	uint32_t	Number of records (0 if not known, i.e. scan)
usRecSize	uint16_t	Size of each record in bytes
usReserved1	uint16_t	Reserved
Data info (must be filled in and passed to create)		
ulDefaultUpdType	uint32_t	Unless otherwise specified, updates will include the following: see PVA_UPDATE_???, see *Table 11: List of UPDATE Bit Definitions* on the next page.
usPosSys	uint16_t	Position coordinate system, see enum *pva_possys_t*, ECEF=1 GEOGRAPHIC=0 ENU=2* BODY=3*
usPosLoc	uint16_t	Position reference location, see enum *pva_posloc_t*, IMU=0, GNSSANT=1, SENSOR1=10
usAttSys	uint16_t	Coordinate system for attitude data, see enum *pva_attsys_t*, RPH=0, WPK=1, BTE=2. **Note**: Only RPH is currently supported.
usAttAlgn	uint16_t	Axes alignment for attitude, see *pva_attalgn_t*, IMU=0, SENS1=10, Currently not used, use PVA_BS_??? commands for sensor orientated data.
usOffsetAvail	uint16_t	1-offset available
usReserved2[5]	uint16_t	Reserved
Data info, datum (optional)		

Variable Name	Type	Description
szDatum[32]	char	Name of datum used. Blank OR "NONE" if not known. Currently, no datum transformation is applied in the software, meaning this parameter is ignored.
szConversion[128]	char	Name of datum conversion, Blank or "AUTO" for auto selection. **Note**: datum conversion is ignored in FileVersion 1
Offset information (40 bytes each, currently on only 1 block)		
fLevBodyX fLevBodyY fLevBodyZ	float	Lever arms values (IMU body axis, m) See also *PVA_LEVER* command for applying offsets during processing
usReserved[2]	uint16_t	Reserved
dBSAtt[3]	double	Boresight attitudes, must match attitude input, currently not used, use *PVA_BS_INIT* command.
usReserved3[1780]	uint16_t	Reserved

* Only used in conjunction with relative updates

Table 11: List of UPDATE Bit Definitions

Name	Hex Value	Description
PVA_UPDATE_NONE	0	No update
PVA_UPDATE_POS	1	Position update to be used and/or available
PVA_UPDATE_VEL	2	Velocity update to be used and/or available
PVA_UPDATE_ATT	4	Attitude update to be used and/or available
PVA_UPDATE_RELATIVE	8	Relative update*
PVA_UPDATE_IGNORE	100	Skip this record
PVA_UPDATE_REJPRE	200	Rejected by pre-filter test
PVA_UPDATE_REJPST	400	Rejected by post-filter test
PVA_UPDATE_TEST	800	Residuals computed (Not used in filter)
PVA_UPDATE_BSVALID	1000	Boresight values are valid in output
PVA_UPDATE_USECMD	10000000	Override record value with passed value
PVA_UPDATE_USEHDR	20000000	Use value from file header
PVA_UPDATE_USEREC	40000000	Use value from each record
PVA_UPDATE_INVALID	80000000	Invalid input or update value (error code)

Table 12: Input data record (pva_inp_type)

Name	Type	Description
dUpdateTime	double	TOW of end of update (s)
dRelStartTime	double	Relative updates only, TOW of start (s), used in rel. only *
usUpdateWeek	uint16_t	GPS week number of UpdateTime
usRelStartWeek	uint16_t	Week number (start, relative only) *
uUpdate	uint32_t	Update/flags type, see *Table 11: List of UPDATE Bit Definitions* on the previous page
szID[16]	char	Optional event 'ID', 15 characters maximum, NULL terminated string
usCheck	uint16_t	Must be 0xC0DE, used as internal check for file validity
usReserved2[3]	uint16_t	Reserved
dPos[3]	double	GEO or ECEF value (lat,lon,hgt or x,y,z) [deg/m]
dAtt[3]	double	Attitude angles RPH, WPK, ECF [deg]. **Note**: Heading and not Yaw, converted to Yaw internally. Currently, only RPH is supported
fVel[3]	float	Velocity (LL or ECF), (e,n,h) or (x,y,z) [m/s]
fPosSD[3]	float	SD values for position, geo(e,n,u)/ecef (x,y,z) (m), as defined by usPosSys in header
fAttSD[3]	float	SD values for attitude, (r,p,h), (w,p,k), or (tx,ty,tz) (deg), as defined by usAttSys in header
fVelSD[3]	float	SD values velocity (e,n,u) for LL or (x,y,z) for ECEF, as defined by usPosSys in header
Off-diagonal correlation coefficients. These are correlation coefficients, defined as: cc = cov[i][j] / (sd_i * sd_j) scale to a signed 16-bit integer using the scale factor 65536.0. cc_int = (int) cc_float * 65536.0		
sCCUpperDiag[36]	int16_t	[pxpy pxpz pxvx pxvy pxvz pxax pxay pxaz] (-1 to 1 after scaling) [pypz pyvx pyvy pyvz pyax pyay pyaz] [pzvx pzvy pzvz pzax pzay pzaz]
ucReserved3[16]	unsigned char	32 bytes reserved

3.3 Inertial Explorer 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.

The forward and reverse loosely and tightly coupled message log files contain all messages output by the processing engine. Inertial Explorer assigns priority levels to all messages generated by the processor and only high priority messages are output to the *Process Window* during GNSS and INS data processing. All messages generated by the processor (regardless of priority) are output to the message log files. These files can be useful in helping to find problems that have not been automatically solved by Inertial Explorer's outlier detection routines.

3.3.2 FL(S)/RL(S)/FT(S)/RT(S)/CT(S) Files

A new output binary format has been created for version 8.70 that reduces the number of trajectory files output from Inertial Explorer. For a copy of the binary struct definitions that define these files please contact support@novatel.com.

Waypoint is aware that there are some products that rely on the old data format from the SBTC/SBIC file. It is still possible to get these files using the *Output | Export to Waypoint Legacy Format* option. See *Export to Waypoint Legacy Format* on page 39 for more details.

Glossary

B

Baseline

Connection between two stations with one or more sessions. Normally, a session and a baseline can be considered the same. However, in some cases there may be more than one session per baseline. This is called a duplicate session baseline and it is plotted yellow on the screen.

C

Check Point

A station with known coordinates, but these coordinates are only used as a check against GrafNet's computed coordinates.

Control Point

See Station or GCP.

G

Ground Control Point (GCP)

A reference station with known latitude, longitude and eight coordinates. The user may also assign horizontal and vertical standard deviations for these values. There can be horizontal, vertical or 3-D points, and there must always be at least one 3-D point or else one horizontal and one vertical point per project.

O

Observation

A raw measurement file collected from a receiver set up over a stationary point. GrafNet only accepts GPB files and, thus, other formats must be converted first. See the table Supported Data Formats for Post-Processing for supported formats. GrafNet also requires single frequency carrier phase data as a minimum, and accepts dual frequency if available. Users wishing to perform code-only processing should use GrafNav.

S

Session

Concurrent period of time between two observation files at two different stations. One of the two stations will be the remote and the other will be the master. The arrow on the screen will be pointing from the master to the remote. The direction is determined by GrafNet in order to form loop closures as well as to minimize the number of legs from a control point. Each session will be processed individually and combined in either a network adjustment or traverse solution. A session can have different statuses and colors depending on whether certain tests passed or failed.

Station

A point where the GPS receiver was setup over and there might be multiple observation files for a single station. However, one set of position values will be produced for each station as a final product of GrafNet. There are several types of stations.

T

Tie Point

Such a point may also be called a loop tie closure and is formed when two or more sessions point to it. Thus, there is a redundant determination at this point.

Traverse Station

This is a point with no tie or control information. It might have two stations connected to it, but one is pointing to it and the other is pointing from it.