OEMSTAR

Application Program Interface (API)

USER GUIDE

OEMSTAR Application Program Interface (API) User Guide

Publication Number: OM-20000131

Revision Level: 1

Revision Date: 2012/2/2

Proprietary Notice

The software described in this document is furnished under a licence agreement or non-disclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or non-disclosure agreement.

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

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

OEMStar and FlexPak are trademarks of NovAtel Inc.

NovAtel® is a registered trademark of NovAtel Inc.

All other brand names are trademarks of their respective holders.

Printed in Canada on recycled paper. Recyclable.

Table of Contents

Tables

Software License

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

1. License: NovAtel Inc. ("NovAtel") grants you a non-exclusive, non-transferable license (not a sale) to, where the Software will be used on NovAtel supplied hardware or in conjunction with other NovAtel supplied software, use the Software with the product(s) as supplied by NovAtel. You agree not to use the Software for any purpose other than the due exercise of the rights and licences hereby agreed to be granted to you.

2. Copyright: NovAtel owns, or has the right to sublicense, all copyright, trade secret, patent and other proprietary rights in the Software and the Software is protected by national copyright laws, international treaty provisions and all other applicable national laws. You must treat the Software like any other copyrighted material except that you may make one copy of the Software solely for backup or archival purposes (one copy may be made for each piece of NovAtel hardware on which it is installed or where used in conjunction with other NovAtel supplied software), the media of said copy shall bear labels showing all trademark and copyright notices that appear on the original copy. You may not copy the product manual or written materials accompanying the Software. No right is conveyed by this Agreement for the use, directly, indirectly, by implication or otherwise by Licensee of the name of NovAtel, or of any trade names or nomenclature used by NovAtel, or any other words or combinations of words proprietary to NovAtel, in connection with this Agreement, without the prior written consent of NovAtel.

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

4. Restrictions: You may not:

(a) copy (other than as provided for in paragraph 2), distribute, transfer, rent, lease, lend, sell or sublicense all or any portion of the Software except in the case of sale of the hardware to a third party;

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

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

(d) transmit the Software over a network, by telephone or electronically using any means (except when downloading a purchased up[grade from the NovAtel Web site); or

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

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

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

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

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

8. Disclaimer and Limitation of Liability:

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

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

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

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

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

Web site: http://www.novatel.com or write to:

NovAtel Inc.
Customer Support Department
1120 - 68 Avenue NE,
Calgary, Alberta, Canada T2E 8S5

Warranty

NovAtel Inc. warrants that its Global Positioning System (GPS) products are free from defects in materials and workmanship, subject to the conditions set forth below, for the following periods of time:

OEMStar Receiver Cards	One (1) Year
OEMStar-Based Receivers	One (1) Year
NovAtel Antennas	One (1) Year
Cables and Accessories	Ninety (90) Days
Software Support	One (1) Year

Date of sale shall mean the date of the invoice to the original customer for the product. NovAtel's responsibility respecting this warranty is solely to product replacement or product repair at an authorized NovAtel location only.

Determination of replacement or repair will be made by NovAtel personnel or by technical personnel expressly authorized by NovAtel for this purpose.

THE FOREGOING WARRANTIES DO NOT EXTEND TO (I) NONCONFORMITIES, DEFECTS OR ERRORS IN THE PRODUCTS DUE TO ACCIDENT, ABUSE, MISUSE OR NEGLIGENT USE OF THE PRODUCTS OR USE IN OTHER THAN A NORMAL AND CUSTOMARY MANNER, ENVIRONMENTAL CONDITIONS NOT CONFORMING TO NOVATEL'S SPECIFICATIONS, OR FAILURE TO FOLLOW PRESCRIBED INSTALLATION, OPERATING AND MAINTENANCE PROCEDURES, (II) DEFECTS, ERRORS OR NONCONFORMITIES IN THE PRODUCTS DUE TO MODIFICATIONS, ALTERATIONS, ADDITIONS OR CHANGES NOT MADE IN ACCORDANCE WITH NOVATEL'S SPECIFICATIONS OR AUTHORIZED BY NOVATEL, (III) NORMAL WEAR AND TEAR, (IV) DAMAGE CAUSED BY FORCE OF NATURE OR ACT OF ANY THIRD PERSON, (V) SHIPPING DAMAGE; OR (VI) SERVICE OR REPAIR OF PRODUCT BY THE DEALER WITHOUT PRIOR WRITTEN CONSENT FROM NOVATEL. IN ADDITION, THE FOREGOING WARRANTIES SHALL NOT APPLY TO PRODUCTS DESIGNATED BY NOVATEL AS BETA SITE TEST SAMPLES, EXPERIMENTAL, DEVELOPMENTAL, PREPRODUCTION, SAMPLE, INCOMPLETE OR OUT OF SPECIFICATION PRODUCTS OR TO RETURNED PRODUCTS IF THE ORIGINAL IDENTIFICATION MARKS HAVE BEEN REMOVED OR ALTERED. THE WARRANTIES AND REMEDIES ARE EXCLUSIVE AND ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, WRITTEN OR ORAL, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE ARE EXCLUDED. NOVATEL SHALL NOT BE LIABLE FOR ANY LOSS, DAMAGE, EXPENSE, OR INJURY ARISING DIRECTLY OR INDIRECTLY OUT OF THE PURCHASE, INSTALLATION, OPERATION, USE OR LICENSING OR PRODUCTS OR SERVICES. IN NO EVENT SHALL NOVATEL BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE DUE TO ANY CAUSE.

There are no user serviceable parts in the GPS receiver and no maintenance is required. When the status code indicates that a unit is faulty, replace with another unit and return the faulty unit to NovAtel Inc.

Before shipping any material to NovAtel or Dealer, please obtain a Return Material Authorization (RMA) number by calling NovAtel Customer Support at 1-800-NOVATEL in North America or 1-403-295-4900 elsewhere.

Once you have obtained an RMA number, you will be advised of proper shipping procedures to return any defective product. When returning any product to NovAtel, please return the defective product in the original packaging to avoid shipping damage.

Foreword

Scope

This document contains sufficient information on working with the Application Program Interface (API) to be able to develop applications for NovAtel's OEMStar receivers. It does not provide specific details of the functions offered by the API, but rather information on building and loading applications. Details of the API functions can be found in the API header file.

Conventions

The conventions used throughout this document are:

< > Text displayed between < and > indicates a variable parameter.

[ ] Text displayed between [and] indicates an optional parameter

✉ This is a notebox that contains important additional information.

In tables where no values are given, such fields should be assumed to be reserved for future use.

Customer Support

If you have any questions or concerns regarding the OEMStar API, use one of the following methods to contact NovAtel Customer Support:

Call the NovAtel Hotline at 1-800-NOVATEL (U.S. & Canada) or +1-403-295-4900 (international)	
Fax: +1-403-295-4901 E-mail: support@novatel.com Web site: http://www.novatel.com	Write: NovAtel Inc. Customer Support Department 1120 - 68 Avenue NE Calgary, AB Canada, T2E 8S5

Chapter 1 Introduction

1.1 Overview

The Application Program Interface (API) allows you to develop specialized C/C++ applications to further extend the functionality of your OEMStar receiver. By using the functions provided by the API, along with the commands and logs already provided by the OEMStar, a wide variety of applications can be created.

1.2 Features

The OEMStar API provides the following features:

- The ability to open physical ports on the receiver to interface with external devices
- Support for three virtual ports, allowing you to directly send commands to and receive logs from the receiver firmware
- Support for multiple tasks, with varying priority levels
- Message queuing functionality
- Support for semaphores and mutexes
- The ability to control many of the receiver's general purpose input/output (GPIO) lines
- Access to receiver time
- Pulse-width modulation control (for receivers that support it)

Refer to the API header file for more information on the features provided by the API.

1.3 Requirements to Build and Run an Application

In addition to the items listed in the API Development Kit (see *Section 1.4*), the following is required to develop and run an application on an OEMStar receiver:

- A NovAtel supported compiler. Currently, *Green Hills Software C/C++* compilers are supported. The *Green Hills Software C/C++* compiler is available in the *Green Hills Multi 2000 Integrated Development Environment*.
- OEMStar receiver loaded with an API enabled software model (See *Section 3.1* on *page 19* for more information about setting up the receiver for loading an application.)
- PC with a serial port and a serial cable to load the application on to the receiver
- OEMStar Firmware Reference Manual

1.3.1 Firmware Compatibility

Firmware support for the API has been designed for backwards compatibility. For example, if the firmware loaded on to the receiver supports version 15 of the API, an application built using version 14 of the API will execute correctly. However, if the application uses a newer version of the API than the firmware supports, the additional or updated functionality provided by the API may not be available.

A description of the changes made to the API since the last version can be found at the beginning of the API header file. To determine what version of the API your firmware supports, log the `APPLICATIONSTATUS` message, which is described further in *Section 4.2* on *page 23*.

Compiler compatibility has been tested successfully up to *Green Hills Multi* V4.2.4. You must run OEMStar firmware version 1.100 or higher.

☒ The API library and header files are named with the firmware version. For example: L6X010100RN0000.api.a.

1.4 Materials Provided

As part of the OEMStar API Development Kit, the following is provided:

- An OEMStar API library file (OEMStar_api.a)
- An OEMStar API header file (OEMStar_api.h)
- One sample project for the Greenhills Multi
- Two Windows command line utility programs, *TOSREC* and *DATABLK*, which format the executable for use with a NovAtel receiver
- NovAtel's *WinLoad* utility for loading the application on to the receiver

Chapter 2 Designing and Building the Application

2.1 Designing the Application

When designing an application for an OEMStar receiver, it is important to have an understanding of some of the key features of the API, which are explained in the sections below.

☒ Review the sample application provided with the API for more information on how to design your application to work with the receiver firmware.

2.1.1 Working with the Virtual Ports

When communicating through one of the physical or virtual ports, the application must be designed to either:

- Read **all** data sent to the port, or
- Configure the port to not transmit data and disable response generation

This is necessary because any incoming data will remain in a buffer until it is read and will not be automatically discarded if more data arrives, resulting in a build up of waiting data.

To disable transmission and response generation at the port, use the `INTERFACEMODE` command with the `NONE` mode for the *txtype* field and `OFF` for the *responses* field. Refer to the *OEMStar Firmware Reference Manual* for more information on this command.

2.1.2 Using the GPIOs

The OEMStar receiver provides LV-TTL general-purpose input/output (GPIO) signals that can be used by your application. The API header file provides more details on the functions available to control and read these GPIOs. The sections below indicate which GPIOs are provided and *Section A.1* on *page 26* gives the electrical specifications for those GPIOs.

2.1.2.1 OEMStar

The following GPIO pins available on the OEMStar:

- GPIO0
- GPIO1
- GPIO3
- PV (output only)

See *Pin-Outs* on page 27 for additional information.

2.1.2.2 FlexPak-G2 with OEMStar

There are two GPIO pins available on the FlexPak-G2 with OEMStar:

- PV
- GPIO1

The *OEMStar Installation and Operation User Manual* provides the location of this pin and details of internal pull-up or pull-down resistors. The electrical specifications for this pin are given in *Table 7* on *page 26*

Note: For OEMStar, Pin 5 = PV line and Pin 6 = GPIO1 line.

2.1.3 Input Parameters

There are two methods that can be used to pass parameters to an application, as discussed in the following sections.

2.1.3.1 Command-Line Entry

The application can be designed to accept a single unsigned, 32-bit parameter, which is then entered as part of the command string for starting the application. This parameter could be used, for example, to set the output serial port used by the application.

For more information about entering a parameter when starting the application, see *Section 3.3.1* on *page 21*.

See *Section 2.1.4* for a note about the use of command-line entry input parameters with auto-start applications.

2.1.3.2 DATABLK Entry

Alternately, the `<SNKey>` field, set when the *DATABLK* utility is run on the executable, can be used to store an input parameter, which would then be read from the `VERSION` log by the application.

For more information about the `<SNKey>` field, see *Section 2.2.3.1* on *page 15*. *Section 4.3* on *page 25* provides more information about where the field is stored in the `VERSION` log.

2.1.4 Auto-Start Applications

An application loaded on to an OEMStar receiver can be set to automatically start whenever the receiver is powered up. This option is set using the `<ComponentEnum #>` field of the *DATABLK* utility, which is described in *Section 2.2.3* on *page 15*.

☒ When an application is configured to automatically start, the input parameter is fixed as `0`, the priority is set to `1`, and the stack size is `10,000`. See *Section 3.3* on *page 21* for more information.

2.2 Building the Application

Building the application in a format that can be loaded on to a receiver is a three step process:

1. Build the application using one of the supported compilers.
2. Convert the binary executable to S-records.
3. Add the S-records necessary for loading the application.

Each of these steps is discussed in the sections that follow.

2.2.1 Building the Application Using a Supported Compiler

2.2.1.1 Project Configuration

With the Greenhills version 4.2.4 compiler, use the build file provided with the sample application as a template. This build file has the necessary settings for a variety of options, including the application format and target, which should not be altered. However, the following can be changed to customize the project, either by directly editing the file or using the interface provided by the compiler.

- The name of the output file
- The files included in the project

2.2.2 Converting Binary to S-Records

When loading the application on to the receiver, the S-record format, which is described in *Section 2.2.2.1*, is used. Therefore, once the application has been built, the resulting binary ELF file must be converted to S-record format. The *TOSREC* utility is provided to complete this conversion.

2.2.2.1 S-Records

The S-record format is an industry standard for encoding programs or data files in a printable form that allows for ease of transfer between devices.

An S-record is an ASCII character string consisting of five fields, in the format shown below.

```
<type><length><address><data...><checksum>
```

The S-record fields all use hexadecimal format, except for the `<type>` field. The fields are described in *Table 1* below.

Table 1: S-Record Fields

Field	Length (Characters)	Description
`<type>`	2	The type of S-record, as described by *Table 2*.
`<length>`	2	The number of character pairs in the record, excluding the `<type>` and `<length>` fields.
`<address>`	4, 6, or 8	The 2, 3, or 4-byte address at which to load the contents of the data field in memory.
`<data>`	variable	Executable code or memory-loadable data.
`<checksum>`	2	The least significant byte of the one's complement of the sum of the values represented by the length, the address, and the data fields.

There are 3 types of S-records used for OEMStar applications, as shown in *Table 2* below.

Table 2: S-Record Types

Type	Description
`S0`	Header record. Used by *WinLoad* to determine how to load the application as described in *Section 2.2.3.2* on *page 17*
`S3`	Data record
`S7`	End-of-file record

Typically, each S-record file consists of one or more header records, followed by one or more data records, concluded with a single end-of-file record.

2.2.2.2 Using TOSREC

The *TOSREC* utility is a Windows command line program. To run the application, enter the following in a command window:

```
tosrec <infile>
```

where `<infile>` is the name of the file to be converted. The name of the S-record file to be generated can be specified using the following option trigger:

```
-o <outfile>
```

where `<outfile>` is the name of the output file.

Examples of command strings to run the utility are given below.

```
tosrec input.elf
tosrec input.elf -o output.hex
```

2.2.3 Adding Information for Loading the Application

The *WinLoad* utility, which is used to load the application on to the receiver, reads information from the input file in order to determine how the application should be loaded. This information is included in a special set of S-records placed at the beginning of the file. Once the application data has been converted to an S-record format, the *DATABLK* utility is used to add these necessary records.

2.2.3.1 Using DATABLK

The *DATABLK* utility is a Windows command line program. To run the application, enter the following string in a command window.

```
datablk <In SREC File> <Out SREC File> <Compress> <Block #>
<ComponentEnum #> <Name> <Version> <SNKey> <Platform> [Compile
Date] [Compile Time]
```

Each of the parameters are described in the table below.

Table 3: DATABLK Parameters

Parameter	Valid Values	Description
`<In SREC File>`	Any	File name of input file.
`<Out SREC File>`	Any	File name of output file.
`<Compress>`	`Compress` or `Raw`	Specifies whether or not the data will be compressed.
`<Block #>`	`0`	The data block in memory in which the application will be loaded into.
`<ComponentEnum #>`	`1` or `5`	The type of application, where `1` specifies a standard user application and `5` specifies an auto-starting user application.
`<Name>`	14 non-null characters or less	A string indicating the name of the application
`<Version>`	14 non-null characters or less	A string indicating the version of the application.
`<SNKey>`	14 non-null characters or less	A string indicating the serial number or key for the application. Can also be used to set an application parameter. See *Section 2.1.3.2* on *page 12* for more details.
<Platform>	14non-null characters or less	A string of comma separated platforms the application is targeting. For OEMStar, valid options are M6XV1G and M6XV1.

Parameter	Valid Values	Description
`[Compile Date]`	Any valid date in the format `yyyy/mmm/dd`, where `mmm` is three letters for the month (eg. JAN)	Optional field to specify the date the application was compiled. If no value is provided, the PC's current date will be used.
`[Compile Time]`	Any valid time in the format `hh:mm:ss`	Optional field to specify the time the application was compiled.If no value is provided, the PC's current time will be used.

An example command string to run the utility is given below.

```
datablk input.hex output.hex raw 2 1 SampleApp 1.00 1234
M6XV1G
```

See *Section 4.3* on *page 25* for more information about how the values entered for the *DATABLK* parameters are used in the receiver logs.

2.2.3.2 Records Used By WinLoad

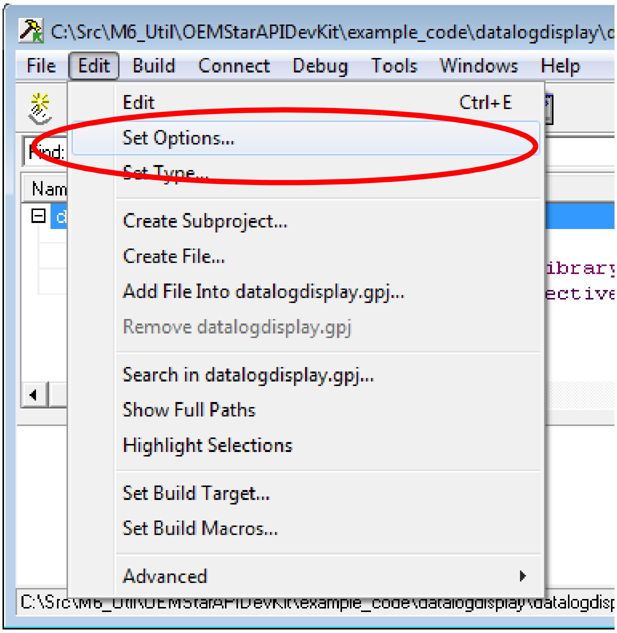
DATABLK adds three header records that are required by the *WinLoad* utility when loading the application. The records provide the following information to *WinLoad*:

- The target platform of the OEMStar receiver (M6XV1 and/or M6V1G)
- The version of the application
- In which data block to load the application (block 0)

2.2.4 Automating Utility Execution

The Greenhills Multi development environment supports execution of specified programs after compilation. This feature can be used to automate execution of TOSREC and DATABLK. By doing so, the binary executable will automatically be converted to S-records and the necessary records for loading will be added. The resultant executable can be loaded directly on to the receiver without further manipulation. Please see below if you are developing with Greenhills Multi.

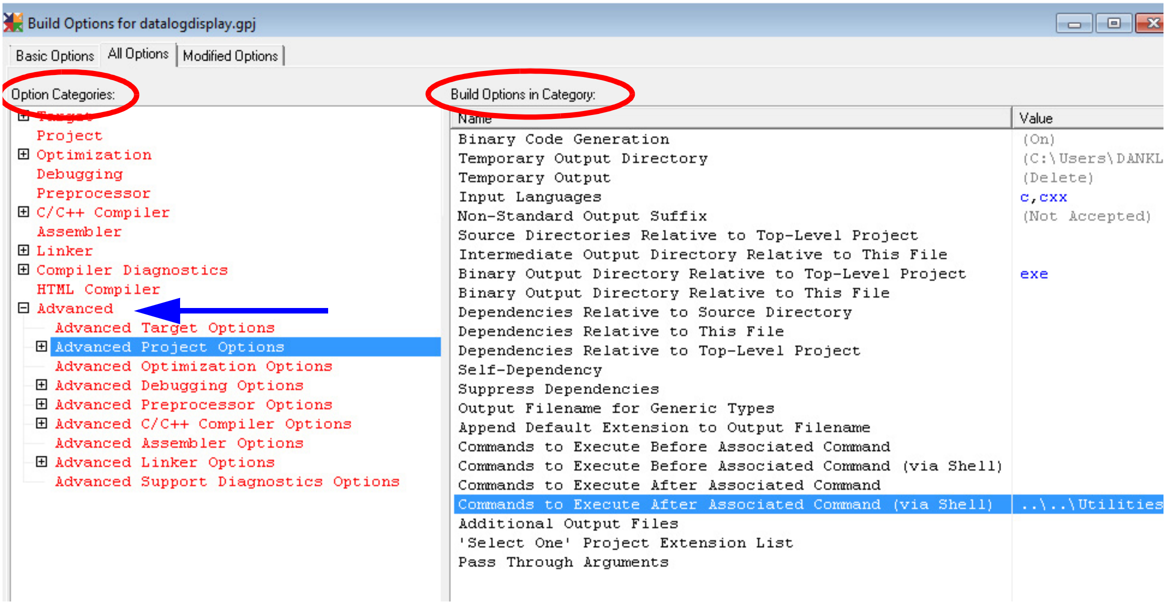
To automate this, with the project open in *Multi*, select *Set Options...* from the *Edit* menu.


C:\Src\M6_Util\OEMStarAPIDevKit\example_code\datalogdisplay\d
File
Edit
Build
Connect
Debug
Tools
Windows
Help
Find:
Nam
Edit Ctrl+E
Set Options...
Set Type...
Create Subproject...
Create File...
Add File Into datalogdisplay.gpj...
Remove datalogdisplay.gpj
Search in datalogdisplay.gpj...
Show Full Paths
Highlight Selections
Set Build Target...
Set Build Macros...
Advanced
C:\Src\M6_Util\OEMStarAPIDevKit\example_code\datalogdisplay\datalogdisp

Expand the **Advance** option under the *Option Categories* on the left side of the window.

Click on **Advanced Project Options** under the *Option Categories* on the left side of the window.

On the right side of the window, under *Build Options in Category*, double-click on **Commands to Execute After Associated Command**.



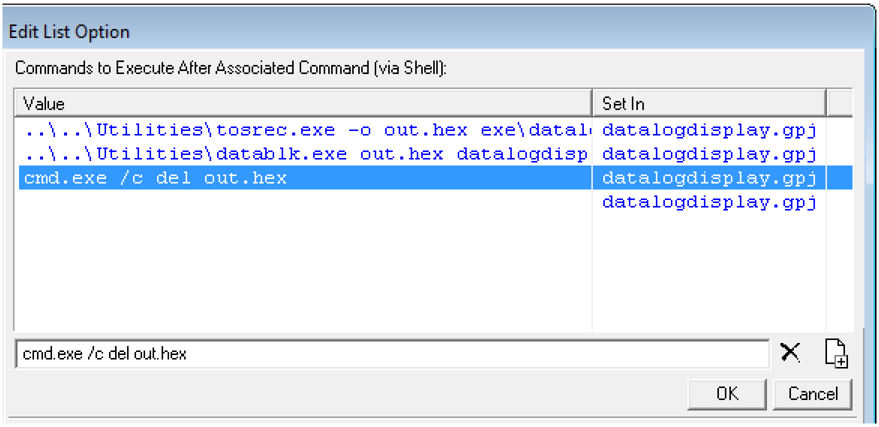
Build Options for datalogdisplay.gpj
Basic Options
All Options
Modified Options
Option Categories:
Build Options in Category:

```
⊞ Target
  Project
⊞ Optimization
  Debugging
  Preprocessor
⊞ C/C++ Compiler
  Assembler
⊞ Linker
⊞ Compiler Diagnostics
  HTML Compiler
⊟ Advanced
    Advanced Target Options
  ⊞ Advanced Project Options
    Advanced Optimization Options
  ⊞ Advanced Debugging Options
  ⊞ Advanced Preprocessor Options
  ⊞ Advanced C/C++ Compiler Options
    Advanced Assembler Options
  ⊞ Advanced Linker Options
    Advanced Support Diagnostics Options
```

Name	Value
Binary Code Generation	(On)
Temporary Output Directory	(C:\Users\DANKL
Temporary Output	(Delete)
Input Languages	c,cxx
Non-Standard Output Suffix	(Not Accepted)
Source Directories Relative to Top-Level Project	
Intermediate Output Directory Relative to This File	
Binary Output Directory Relative to Top-Level Project	exe
Binary Output Directory Relative to This File	
Dependencies Relative to Source Directory	
Dependencies Relative to This File	
Dependencies Relative to Top-Level Project	
Self-Dependency	
Suppress Dependencies	
Output Filename for Generic Types	
Append Default Extension to Output Filename	
Commands to Execute Before Associated Command	
Commands to Execute Before Associated Command (via Shell)	
Commands to Execute After Associated Command	
Commands to Execute After Associated Command (via Shell)	..\..\Utilities
Additional Output Files	
'Select One' Project Extension List	
Pass Through Arguments	

In the *Commands to Execute After Associated Command* box, enter the command strings to run the utilities.

Press *OK* to save the changes.



Commands to Execute After Associated Command (via Shell):

Value	Set In
..\..\Utilities\tosrec.exe -o out.hex exe\datal	datalogdisplay.gpj
..\..\Utilities\datablk.exe out.hex datalogdisp	datalogdisplay.gpj
cmd.exe /c del out.hex	datalogdisplay.gpj
	datalogdisplay.gpj

cmd.exe /c del out.hex

The next time the application is built, the utilities will automatically be executed as specified.

Chapter 3 Loading and Controlling the Application

Once the application has been built and converted to S-record format, with the necessary S-records added, the application can be loaded on to the receiver.

A Windows-based utility named *WinLoad* has been created to assist with loading the firmware.

3.1 Setting up the Receiver

In order to load and run an application, the receiver must have:

- A model with the API option enabled, and
- Version L6X010101RN0000 or higher of firmware loaded

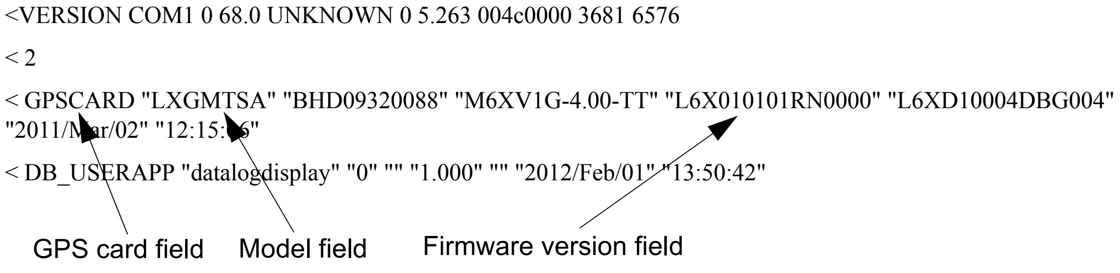
The following sections provide information on how to determine if your receiver meets these criteria and how to update it if it does not.

3.1.1 Determining the Current Model and Firmware Version

To determine the current model and firmware version of the receiver, read the `VERSION` log. To do so, send the following command to the receiver.

```
LOG VERSION
```

Read the output provided, specifically the 1st and 4th fields after the word `GPSCARD`. The first field provides the model and the fourth field indicates the version of firmware loaded on the receiver. In the example below, the model is "LXGMTSA" and the firmware version field shows "L6X010101RN0000".



```
<VERSION COM1 0 68.0 UNKNOWN 0 5.263 004c0000 3681 6576

< 2

< GPSCARD "LXGMTSA" "BHD09320088" "M6XV1G-4.00-TT" "L6X010101RN0000" "L6XD10004DBG004"
"2011/Mar/02" "12:15:06"

< DB_USERAPP "datalogdisplay" "0" "" "1.000" "" "2012/Feb/01" "13:50:42"
```

GPS card field
Model field
Firmware version field

In this example, the model field ends with an "`A`" and the firmware version field reads "L6X010101RN0000" therefore, an application can be loaded on to the receiver. If an update to the firmware and/or model is needed, instructions are provided in *Section 3.1.2* and *Section 3.1.3*.

3.1.2 Updating the Firmware

To update the firmware to a version that supports the API, obtain the following from NovAtel Customer Support.

- The firmware update file, with version 1.100 or higher
- An update authorization code

WinLoad is also required to load the firmware on to the receiver and is included with the API package. Follow the procedure given in the `HowTo.txt` file provided with the update file to upgrade the firmware.

3.1.3 Authorizing a Model with the API Option

To authorize a model with the API option enabled, contact NovAtel Customer Support to obtain the necessary authorization code and use the `AUTH` command to add the code to the receiver. Refer to the *OEMStar Firmware Reference Manual* for more information on this command.

3.2 Loading the Application

Once the receiver is set up with the necessary model and firmware version, the application can be loaded in to non-volatile memory on the receiver using release 102 or higher of the *WinLoad* utility. This utility is provided as part of the API package or can be obtained from NovAtel Customer Support.

By reading the S-records added using the *DATABLK* utility, *WinLoad* automatically knows where and how to place the application in memory so it does not interfere with the operation of the main receiver firmware. Therefore, loading the application can be done using the same method as that used when loading GPS firmware, with the following exceptions:

- The application `HEX` file should be selected, rather than a standard firmware `HEX` file.
- An authorization code is not needed and, therefore, you will not be prompted for one.

For more information on using *WinLoad* to load the application, follow the procedure given in the *OEMStar Installation and Operation User Manual*, making adjustments for the selection of the application file, rather than the firmware file, and the absent authorization code prompt.

☒ Only one application can be loaded on to the receiver at any one time. However, functions are provided to allow for multiple tasks running within the application. Please see the provided library files for more information.

3.3 Controlling the Application

The operation of the application can be controlled by using the APPLICATION command as discussed below.

3.3.1 Starting the Application

Once the application has been loaded on to the receiver, enter the following command string to start the application.

```
application start <parameter> <priority> <stack>
```

The values that can be entered when starting an application are described in the table below.

Table 4: APPLICATION START Parameters

Parameter	Valid Values	Description
<parameter>	Any ulong value	Optional field to specify an input parameter for the application. If a value is not specified, the default of 0 is used.
<priority>	Any long value from 0 to 16	Optional field to specify the priority of the application in relation to system tasks. See the API header file for details. In order to specify the priority, a value must be entered for the <parameter> field as well. If a value is not specified, the default of 1 is used.
<stack>	Any long value from 200 to 20000	Optional field to specify the size of the stack to be used by the application in bytes. In order to specify the stack size, a value must be entered for both the <parameter> and <priority> fields as well. If a value is not specified, the default of 10000 is used.

3.3.1.1 Auto-Start Applications

If 5 was entered for the <ComponentEnum #> field when running the *DATABLK* utility, the application will be set to auto-start. Therefore, when the receiver is first powered, the application will begin running.

An alternate method for setting an application to automatically start is by using the SAVECONFIG command after the application has been started. When this is done, the application will automatically start whenever the receiver is powered up as long as the saved receiver configuration is present in memory. However, as soon as the FRESET command is issued, the configuration will be lost and the application will not automatically start. When an application is configured to auto-start using the method described in *Section 2.1.4* on page 12, it will not be affected by the state of the receiver configuration.

3.3.2 Stopping the Application

To stop the application, enter the following command.

```
application stop
```

This command can be used to stop either standard or auto-start applications.

3.3.3 Removing the Application

To remove the application from non-volatile memory, enter the following command.

```
application remove
```

⌧ The application is not recoverable when removed from non-volatile memory, so be sure to use this command with care.

Chapter 4 Additional Information

The following sections provide additional information that may be useful when working with OEMStar applications.

4.1 Determining the Version of the Loaded Application

When an application is loaded on to the receiver, the `VERSION` log provides information about the application. When an application is loaded, an additional entry is displayed in the `VERSION` log, with the `type` field showing `DB_USERAPP` for a standard application or `DB_USERAPPAUTO` for an auto-start application. All the parameters given in that entry apply to the application loaded on to the receiver. An example of the log with a `DB_USERAPP` entry is shown below.

```
<VERSION COM1 0 68.0 UNKNOWN 0 5.263 004c0000 3681 6576

< 2

< GPSCARD "LXGMTSA" "BHD09320088" "M6XV1G-4.00-TT" "L6X010101RN0000"
"L6XD10004DBG004" "2011/Mar/02" "12:15:06"

< DB_USERAPP "datalogdisplay" "0" "" "1.000" "" "2012/Feb/01" "13:50:42"
```

For information on how to capture the `VERSION` log or the fields it contains, refer to the *OEMStar Firmware Reference Manual*. *Section 4.3* on *page 25* provides information on how some of the `VERSION` log fields are set by the *DATABLK* utility.

4.2 Logging the Application Status

A log has been created to capture the details of any application loaded on to the receiver. The details of this log follow. All formats and standard fields, such as the header, are explained further in the *OEMStar Firmware Reference Manual*. Many of the fields are set to values entered for use by the *DATABLK* utility, as described in *Section 4.3* on *page 25*.

☒ The time indicated in the log header is the time when the application status was last changed, for example, the time when the application was started or stopped.

APPLICATIONSTATUS API Application Status Information

Log Type: Asynch **Message ID: 520**

Field	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	header	Log header		H	0
2	api version	The version of the API that the currently loaded firmware supports	Ulong	4	H
3	running	Flag indicating whether the application is currently running, where 0 = FALSE 1 = TRUE	Enum	4	H+4
4	base address	The base address in RAM the application is running from Valid only when the application is running	Ulong	4	H+8
5	size	The number of bytes in RAM the application is using Valid only when the application is running	Ulong	4	H+12
6	name	The name of the application	Char[16]	16	H+16
7	version	The version of the application	Char[16]	16	H+32
8	compile date	The date the application was compiled In the format `yyyy/mmm/dd`, where `mmm` is three letters for the month (eg. `JAN`)	Char[12]	12	H+48
9	compile time	The time the application was compiled In the format `hh:mm:ss`	Char[12]	12	H+60
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
11	[CR][LF]	Sentence terminator (ASCII only)			

Recommended Input:

```
LOG APPLICATIONSTATUSA
```

ASCII Example:

```
#APPLICATIONSTATUSA,COM1,0,75.0,UNKNOWN,0,1.814,004c0000,3314,6576;1,FALSE,0000
0000,00000000,"datalogdisplay","1.000","2012/Feb/01","13:50:42"*3eb207f8
```

4.3 DATABLK Parameters Used in Logs

Many of the fields captured in the `APPLICATIONSTATUS` log and the `DB_USERAPP` or `DB_USERAPPAUTO` entry of the `VERSION` log are set to the values entered for parameters used by the *DATABLK* utility. The table below provides a list of these parameters and the matching log fields. For more information about the DATABLK utility, see *Section 2.2.3.1* on *page 15*.

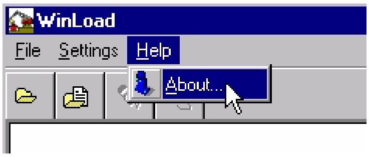
Table 5: DATABLK Parameters Used in Logs

DATABLK Parameter	Matching Log Field	
	VERSION[a]	APPLICATIONSTATUS
`<Name>`	model	name
`<SNKey>`	psn	N/A
`<Version>`	sw version	version
`<Compile Date>`	comp date	compile date
`<Compile Time>`	comp time	compile time

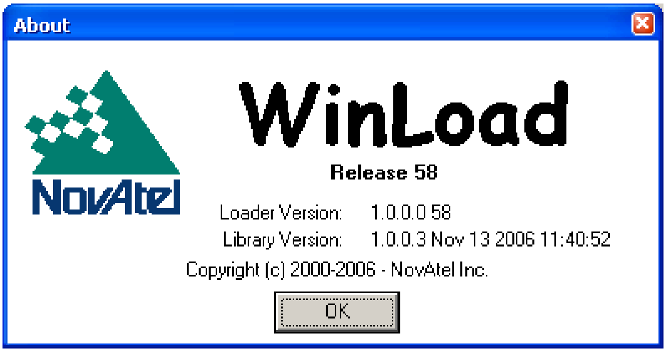
[a] Only valid for the `DB_USERAPP` or `DB_USERAPPAUTO` entry in the log.

4.4 Determining the Version of WinLoad

To load an application on to the receiver, release 102 or higher of *WinLoad* must be used. To determine the version of *WinLoad* you currently have, run the application and select *About...* from the *Help* menu.



The version is shown in the *Loader Version* field in the *About* dialog box.


About

WinLoad

Release 58

Loader Version: 1.0.0.0 58

Library Version: 1.0.0.3 Nov 13 2006 11:40:52

Copyright (c) 2000-2006 - NovAtel Inc.

OK

Appendix A Technical Specifications

A.1 GPIO Electrical Specifications

A.1.1 OEMStar

Table 6: OEMStar GPIO Electrical Specifications

	Low		High	
	Voltage	Current	Voltage	Current
Input	$0 < V_{IL} < 0.8$ VDC	$I_{IL} = -0.1$ mA	$2.0 < V_{IH} < 3.6$VDC	$I_{IH} \sim 0$ mA
Output	$V_{OL} < 0.4$ VDC	$I_{OLMAX} = 8$ mA	$V_{OH} > 2.7$ VDC	$I_{OHMAX} = -8$ mA

A.1.2 FlexPak-G2 with OEMStar

Table 7: FlexPak-G2 with OEMStar GPIO Electrical Specifications

	Low		High	
	Voltage	Current	Voltage	Current
Input	$0 < V_{IL} < 0.8$ VDC	$I_{IL} = -0.1$ mA	$2.0 < V_{IH} < 3.6$VDC	$I_{IH} \sim 0$ mA
Output	$V_{OL} < 0.4$ VDC	$I_{OLMAX} = 8$ mA	$V_{OH} > 2.7$ VDC	$I_{OHMAX} = -8$ mA

A.1.3 Pin-Outs

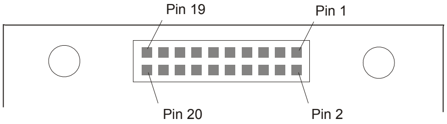


Figure 1: Top-view of 20-Pin Connector on the OEMStar

Signal	Behavior [a]	Descriptions	Pin
GPIO0	Input / Output	User-configurable I/O pin	7
GPIO1	Input / Output	User-configurable I/O pin	8
GPIO3	Input / Output	User-configurable I/O pin	17
PV	Output	Output shows a good solution, or valid GPS position, when high	20

[a] The VIN and LNA_PWR inputs are protected against damage or latch-up due to ESD by bi-directional Transient Voltage Suppressor (TVS) devices. GPIO lines are protected by undirectional TVS devices and series resistors.

Index

Recyclable

♻ ♻

Printed in Canada on recycled paper