MarshallSoft GPS Component

Users Manual

(MGC_USR)

Version 2.2

June 8, 2011

This software is provided as-is.
There are no warranties, expressed or implied.

MarshallSoft Computing, Inc.
Post Office Box 4543
Huntsville AL 35815 USA

Voice : 1.256.881.4630
email : info@marshallsoft.com
web : www.marshallsoft.com

MARSHALLSOFT is a registered trademark of MarshallSoft Computing.

TABLE OF CONTENTS

1 Introduction

The **MarshallSoft GPS Component (MGC)** is a 32-bit dynamic link library (DLL) which reads and decodes standard GPS (Global Positioning System) NMEA (National Marine Electronics Association) 183 sentences from the RS232 serial port as well as computes great circle distances and bearings.

The **MGC** library can be called from any application capable of calling Windows API functions, including those written in C/C++, .NET, Delphi, Visual Basic, MS Access, MS Excel, Fortran, COBOL, PowerBuilder, Visual FoxPro, Power Basic, dBase, Xbase++, etc. We have versions of the **MarshallSoft GPS Component SDK** for C/C++ (MGC4C) and Visual Basic (MGC4VB).

Purchase a developer license for one software development language and use it with all others. All versions of MGC use the same DLL (MGC32.DLL), however, the examples provided for each version are written and tested for the specified computer development language.

We also have declaration files and example programs for a few other languages. The **MarshallSoft GPS Component** DLL (MGC32.DLL) runs under all versions of Windows (Windows 98, Windows ME, Windows 2000, Windows 2003, Windows NT, Windows XP, and Vista).

The <u>latest</u> <u>version</u> of the **MarshallSoft GPS Component** software can be found online at:

http://www.marshallsoft.com/gps-communication-library.htm

This manual applies to the **MarshallSoft GPS Component (MGC)** component library for all supported languages

1.1 MarshallSoft GPS Component Features

Some of the many features of the **MarshallSoft GPS Component (MGC)** library toolkit are:

- Uses the Windows Win32/Win64 API for serial port input. No special Windows driver is required.
- Runs as a background thread unattended. MGC is fully thread safe.
- The most current GPS data is always available on demand.
- Includes support functions such as **mgcGreatCircle.**
- Can read any GPS NMEA 183 sentence, extracting each field
- Can read and decode GPRMC, GPGGA, GPGLL, GPGSV, GPVTG, GPBOD, GPWPL, and GPGSA navigation sentences. Additional sentence types can easily be added.
- Can graphically display latitude and longitude.
- Can compute distances and bearings.
- Provides ability to convert units.
- Works with virtual serial ports such as Bluetooth serial.
- Work with USB ports that have a "USB to Serial Port" converter cable
- Can be called from any application program capable of calling the Windows Win32 API, including C/C++, .NET, Visual Basic, VB.NET, PowerBASIC, Delphi, Delphi .NET, Visual FoxPro, Visual dBase, Xbase++, Fortran, COBOL, etc.
- Also can be called from any application program capable of calling the Windows Win64 API such as Visual Studio (C/++ & Visual Basic).
- **Free** technical support and updates for one year.
- Royalty free distribution with your compiled application.
- Evaluation versions are fully functional. No unlock code is required.

1.2 Documentation Set

The complete set of documentation consists of three manuals in two formats. This is the second manual (MGC_USR) in the set.

- MGC_4x Programmer's Manual `(MGC_4x.PDF)`
- MGC User's Manual `(MGC_USR.PDF)`
- MGC Reference Manual `(MGC_REF.PDF)`

The MGC_4x Programmer's Manual is the language specific manual. All language dependent programming issues such as compiling, compilers and example programs are discussed in this manual. Other language specific manuals are as follows:

```
[NAME]      [DESCRIPTION]
MGC_4C   :  MGC Programmer's Manual for C/C++
MGC_4D   :  MGC Programmer's Manual for Delphi        [not released]
MGC_4VB  :  MGC Programmer's Manual for Visual Basic
MGC_4PB  :  MGC Programmer's Manual for PowerBASIC    [not released]
MGC_4FP  :  MGC Programmer's Manual for Visual FoxPro [not released]
MGC_4DB  :  MGC Programmer's Manual for Visual dBase  [not released]
MGC_4XB  :  MGC Programmer's Manual for Xbase++       [not released]
```

The MGC User's Manual (MGC_USR) discusses GPS (Global Positioning System) fundamentals as well as language independent programming issues such as application notes, purchasing and licensing. Read this manual after reading the MGC Programmer's Manual.

The MGC Reference Manual (MGC_REF) contains details on each individual MGC function and includes a list of error codes.

1.3 Technical Support

We want you to be successful in developing software applications using our **MarshallSoft GPS Component** (MGC) library! We are committed to providing the best, most robust software development toolkit that we can. If you have any suggestions for enhancements or comments, please let us know.

If you are having a problem using the MarshallSoft GPS Component, refer to Section 8.0 "Problems". If you still cannot resolve your problem, email us at

> support@marshallsoft.com

To avoid having your email deleted by our Spam scanners, begin the subject with MGC. Zip up any attachments and send plain ASCII text email only.

Support can be reached at +1-256-881-4630 between 7:00 AM and 7:00 PM CST Monday through Friday.

The latest versions of our products are available on our web site at

> http://www.marshallsoft.com

and on our anonymous FTP sites at

> ftp://ftp.marshallsoft.com/pub

Registered users with a current license (less than 13 months old) can update to the latest DLL's at

> http://www.marshallsoft.com/oem.htm

1.4 How to Purchase

The developer's license for the **MarshallSoft GPS Component SDK (MGC**) may be obtained for $115 (USD) for electronic delivery. **MGC** with ANSI C source code for the DLL is available for $195 (USD). This price is good for one year from the release date.

The fastest and easiest way to purchase our software is on our web site at

http://www.marshallsoft.com/order.htm

You can also order by completing INVOICE.TXT (contained within the **MGC** zip file) and emailing (info@marshallsoft.com), mailing (see our address at top), or faxing (+1.256.880.0925) it to us.

Multiple copy discounts (3 or more) and site licenses are available. Please call for details.

We accept American Express, VISA, MasterCard, Discover, checks in US dollars drawn on a US bank, International Postal Money Orders, Paypal from verified accounts, and purchase orders (POs) within the USA from recognized US schools and companies listed in Dun & Bradstreet.

For credit card orders, be sure to include the account number, the expiration date, the exact name on the card, and the complete card billing address (the address to which the credit card bill is mailed- <u>not</u> the banks). Please include the Card Verification Code (last 3 numbers printed on the back of Visa, MasterCard and Discover cards, or the 4 numbers of the front of American Express cards.) The cardholder's signature is required on faxed orders

Print the file INVOICE.TXT if a "Pro Forma" invoice is needed. The registered package includes:

- Win32 MGC Library without the evaluation reminder popup window.
- Free telephone and email support for one year.
- Free downloadable updates to the registered DLL for one year.

Note that the purchased **MGC** DLL never expires; only the ability to download new releases is limited.

1.4.1 Source Code

There are two ways to order Source Code for the **MarshallSoft GPS Component** software.

(1) ANSI C source code can be ordered at the same time as the Developer's License for $195 (for both).

(2) Source code can be ordered within one year of purchasing a Developer's License for $100. After one year, a Developer's License update ($30, $55 or $75) must be purchased prior to purchasing the source code.

1.4.2 Academic Discount

We offer an "academic price" of 40% off the normal price for prepaid email orders to faculty and <u>full</u> time students currently enrolled in any accredited high school, college, or university. To qualify for the discount, your school must have a web site and you must have an email address at your school.

When ordering by phone, ask for the "academic discount". If ordering on our web site, enter "student at" (or "faculty at") and your schools web site address (URL) in the comments field of the order form. Your order will be sent to your email address at your school.

This offer is not retroactive and cannot be used with any other discount. Products bought with academic pricing are for **<u>educational use only</u>**. They can <u>not</u> be used for any commercial purpose.

1.5 Updates

When a developer license is purchased for the **MarshallSoft GP Component SDK,** the developer will receive a set of registered DLLs plus a license file (MGCxxxx.LIC) that can be used to update the registered DLL's (does not include source code) for a period of <u>one year</u> from purchase.

http://www.marshallsoft.com/oem.htm

After one year, the developer license must be updated to be able to download updates to the library. A developer license can be updated for $30 if ordered within <u>one year</u> of the original purchase (or previous update). After one year, licenses can be updated for $55 and after 3 years licenses can be updated for $75.

These update prices do <u>not</u> include source code.

After one year, the developer license must be updated to be able to download updates. The developer license can be updated for $30 if ordered within <u>one year</u> from the original purchase (or previous update). After one year, licenses can be updated for $55 ($75 after 3 years).

Source code previously purchased may be <u>updated</u> for $50 in addition to the cost of the update ($30, $55 or $75).

Note that the registered MGC DLL does **not** expire.

Also see file UPDATES.TXT.

1.6 Customer ID

The Customer ID is the 5 or 6 digits following the product name (**MGC**) in the license file. For example, customer 12345 will receive license file **MGC12345.LIC**. Provide the Customer ID when contacting us for technical support (MGC4C 12345).

1.7 License File

When a developer license is purchased, the developer will receive a license file named MGCxxxx.LIC, where "xxxx" is the 4 or 5 digit customer ID. The license file is an encrypted binary file used for updating **MGC** as explained in section 1.5 "Updates". The license file is required in order to create (or update) the registered DLL's.

1.8 Distribution

In order to run your application (that calls **MGC** functions) on another computer, the file MGC32.DLL must be copied to the Windows directory of the other computer. The Windows directory is normally \WINDOWS for Windows 98/ME/XP/2003/Vista and \WINNT for Windows NT/2000. Do not attempt to "register" the DLL with Windows; it is not necessary.

2 Library Overview

2.1 Keycode

When you register the **MarshallSoft GPS Component SDK**, you will receive a new DLL and a keycode for your DLL. Pass this keycode as the second argument to **mgcAttach**. The keycode will be found in the file named "**KEYCODE**". The keycode for the evaluation version is 0. The keycode for the registered version will be a unique 9 or 10 digit number. Note: Your keycode is NOT your Customer ID/Registration number. . After registering, it is best to remove the evaluation version of MGC32.DLL from the Windows search path.

If you get an error message (value -74) when calling **mgcAttach**, it means that the keycode in your application does not match the keycode in the DLL

2.2 Dynamic Link Libraries

The **MarshallSoft GPS Component SDK** is implemented as a Win32 dynamic link library (MGC32.DLL). A DLL is characterized by the fact that it need not be loaded until required by an application program and that only one copy of the DLL is necessary regardless of the number of application programs that use it. Contrast this to a static library that is bound at link time to each and every application that uses it.

2.3 GUI and Console Mode

MGC functions can be called from WIN32 console mode programs as well as GUI programs. A "console mode" program is a Windows 98/Me/NT/2000/XP/2003/Vista WIN32 command line program running in a command window. Although console mode programs look like DOS programs, they are WIN32 programs that have access to the entire Windows address space.

2.4 Using the Library

The first **MGC** function that should be called is **mgcAttach**, which initializes the **MGC** library and allocates necessary resources. **mgcAttach** is typically called in the initialization section of the application and should be called just once. The function **mgcRelease** should be called before exiting your application.

The best way to get familiar with **MGC** is to try out one of the example programs. The example programs are described in the MGC_4x Programmer's Manual. The "x" in MGC_4x specifies the host language such as C for C/C++, VB for Visual Basic, etc. the example source is written in.

```
[NAME]        [DESCRIPTION]
MGC_4C    :  MGC Programmer's Manual for C/C++
MGC_4D    :  MGC Programmer's Manual for Delphi
MGC_4VB   :  MGC Programmer's Manual for Visual Basic
```

2.5 Interfacing MGC with Other Languages

The MarshallSoft GPS Component DLL (MGC32.DLL) can also be used with any application written in any language capable of calling the Windows 32-bit API (98/Me,Vista/ NT/2000/2003/XP).

Once you have purchased one language version of **the MarshallSoft GPS Component (MGC)** Library SDK, you can use it with all other supported languages. Supported (Win32) languages are C/C++, Delphi and Visual Basic.

See LANGUAGES.TXT in the APPS directory for more information regarding using **MGC** with multiple languages.

Declaration files have been defined for the following software languages:

```
[LANGUAGE]              [FILE]
C/C++                   MGC.H
Visual Basic            MGC.BAS
VBA (Excel, Access)     MGC.BAS
PowerBASIC              MGC.PBI
Borland Delphi          MGC.PAS
Visual FoxPro           MGC.FOX
Visual dBase            MGC.CC
Alaska Xbase++          MGC.CH
```

Additional declaration files will be added. Give us a call if you need a declaration not listed above.

2.6 Bluetooth

Bluetooth is a short range radio technology that is used to communicate between devices located within a small area (typically less than 30 feet). Most Bluetooth adaptors use an USB port. Examples include the LinkSys USBBT-100, the D-Link DBT-120, the Belkin Bluetooth USB Adapter as well as many others.

Most Bluetooth adapters include serial port emulation. The Bluetooth adapter comes with a Windows device driver that emulates a serial port. After installing your Bluetooth adapter, a new virtual serial port will become available. Start the Windows Device Manager, and then look under "Ports".

Some Bluetooth serial port implementations use an "inbound" and an "outbound" serial port. The Bluetooth device that initiates the serial connection between devices uses the outbound port (for serial transfer in both directions) while the other device uses it's inbound port (also for both directions). Note that an application will never use both inbound and outbound ports at the same time.

A Bluetooth enabled computer can communicate with a Bluetooth enabled GPS receiver, such as the Teletype GPS and many others. The GPS receiver transmits NMEA 183 sentences over its Bluetooth serial port that can then be received by a Bluetooth enabled computer.

2.7 USB Ports

MGC will work with USB ports that have a "USB to Serial Port" converter cable (such as the D-Link DSB-S25) installed. After installing the USB/serial driver that comes with the USB converter, a virtual RS232 port will created.

Connect the serial GPS receiver to the USB/serial cable and everything should work the same as a real RS232 serial port.

3 GPS Background

The following provides a high level background for discussion of GPS.

3.1 What is GPS ?

The Global Positioning System (GPS) is a constellation of 24 satellites that orbits the earth and used for navigation by anyone with a GPS receiver.

The GPS satellites were placed into orbit by the U.S. Department of Defense for military applications. In the 1980s, the US government made the GPS system available for civilian use. The accuracy of the GPS signal was intentionally degraded (known as Selective Availability) for non-US military users, but has recently been removed (May 2nd, 2000).

There are many good sites on the internet dealing with GPS. Some good ones are

http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html
http://www.aero.org/education/primers/gps/

3.2 How Does GPS Works

The 24 GPS satellites (21 active plus 3 spares) are in orbit at 10,600 miles above the surface of the earth. The satellites are spaced such that four satellites will be above the horizon when viewed from any point on the surface of the earth.

GPS satellites circle the earth twice each day. The GPS receiver compares the time that a signal was transmitted by a satellite with the time that it was received. The time difference determines the distance between the GPS receiver and the satellite.

A GPS receiver must be locked to the signal of at least three satellites in order to calculate a position (by spherical triangulation) on the surface of the earth, giving latitude and longitude of the GPS receiver. With four or more satellites in view, the receiver can also determine the altitude.

3.3 WGS-84

WGS-84 (World Geodetic System 1984) is a global reference frame for the earth. It was accepted in March 1989 as the standard world reference frame by the International Civil Aviation Organization (ICAO).

Most GPS receivers today model the earth as a geode as defined by WGS-84 (World Geodetic System 1984).

Earth's radius as defined by WGS-84 is as follows:

```
Equatorial radius = 6,378,137.0 m
Polar radius      = 6,356,752.3142 m
```

For more information on WGS-84, refer to:

http://earth-info.nga.mil/GandG/wgs84/index.html

4 GPS Units

4.1 Latitude and Longitude

A great circle is the intersection of a sphere with a plane that passes through the center of the sphere. Note that the shortest distance (on the surface of the sphere) between two points is always part of a great circle.

Latitude and longitude are spherical coordinates on the surface of the earth.

Lines of latitude, also called parallels, are horizontal to each other and run east and west. Latitude is measured north (+) or south (-) of the Equator, resulting in a range of -90 to +90 degrees. The Equator is the parallel (also a great circle) that divides the globe into two halves, the Northern and Southern Hemispheres. Any point on the globe below the equator has a negative value for latitude, while points above the equator have positive latitude values.

Lines of longitude, also called meridians, are great circles that run between the North Pole and the South Pole. Thus, lines of latitude and lines of longitude always meet at right angles on the surface of the earth. Longitude is measured east (+) or west (-) of the prime meridian which passes through Greenwich, England. Longitude is measured east or west from 0 to 180 degrees.

Both latitude and longitude are measured in degrees (360 degrees in a circle). Each degree can be subdivided into 60 minutes, and each minute can be subdivided in 60 seconds. In the NMEA Sentence format, latitude and longitude are measured in degrees, minutes, and thousandths of minutes. In the **MarshallSoft GPS Component (MGC)**, latitude and longitude are measured in integral degrees, integral minutes, and integral units of thousandths of a minute.

Every location on the surface of the earth has unique latitude and longitude coordinates. Determining the latitude and longitude of your location is the primary purpose of a GPS receiver.

4.2 Altitude

Altitude is the distance above the WGS-84 reference geode, measured in feet or meters (default). Altitude measurements in GPS are typically not as accurate as latitude and longitude measurements.

4.3 Ground Distance

Ground distance is the distance along a great circle (arc length) between two points on the WGS-84 reference geode, measured in feet or meters (default).

4.4 Calculations

There are four common ways that latitude and longitude values are expressed:

4.4.1 (deg, min, sec) Form

Latitude and Longitude are specified by (X, Y, Z) where X is integral degrees, Y is integral minutes, and Z is integral seconds.

Example: 34 deg 30 min 21 sec = (34, 30, 21)

4.4.2 (deg, min, min/1000) Form (reported by NMEA 183 GPS)

Latitude and Longitude are specified by (X, Y.Z) where X is integral degrees, Y is integral minutes, and Z is in units of thousandths of a minute (1000 units = 1 minute).

Example: 34 deg 30.350 min = (34, 30.350)

4.4.3 Decimal degree Form

Latitude and Longitude are specified by decimal degrees

Example: 34.50583 degrees

4.4.4 Coded Integer Form

Latitude and Longitude are specified in units of thousandths of a minute. That is, 1000 units equals one minute and 60000 units equals one degree.

Example:

34.50583 deg = (34 deg, 30.350 min) = (34 deg, 30 min, 21 sec)   = 2070350 (coded integer) since

```
    2070350 = 60000 * 34
            +  1000 * 30
            +    350
```

Coded integer values can be either positive (north latitude and east longitude) or negative (south latitude and west longitude).

See the MGC conversion routines **mgcDecodeDeg**, **mgcDecodeMin**, **mgcDecodeSec**, **mgcDecodeTho**, **mgcEncodeDMS**, and **mgcEncodeDMT**.

Also see the COMPUTE and CONVERT example programs.

5 Using the MarshallSoft GPS Component Software (MGC)

5.1 Coordinates Used in MGC.

5.1.1 Latitude and Longitude

It is advantageous to represent latitude and longitude values as integers rather than as floating point, real, or packed decimal numbers. This allows the MGC DLL (MGC32.DLL) functions to be called by any application regardless of the type of floating point (real) numbers supported by the language.

Latitude and longitude coordinates are specified in the "Coded Integer Form" as described in the section above. Refer to the CONVERT example program for a demonstration of converting between the various forms.

5.1.2 Altitude

Altitude in **MGC** can be returned in units of feet (F) or meters (M), depending on the value (*) set when calling

```
mgcSetInteger(MGC_SET_ALTITUDE_UNIT, * )
```

The default is meters. Refer to the MGC Reference Manual (MGC_REF) for more details.

5.1.3 Ground Distance

Ground distance in **MGC** can be returned in units of feet (F), meters (M), kilometers (KM), statue miles (SM), or nautical miles (NM), depending on the value (*) set when calling

```
mgcSetInteger(MGC_SET_DISTANCE_UNIT, * )
```

The default is meters. Refer to the MGC Reference Manual (MGC_REF) for more details.

5.2 How the MGC Software Works.

The **MarshallSoft GPS Component (MGC)** software (implemented in MGC32.DLL) reads NMEA sentences from the selected serial port.

Calling **mgcOpen** in the MGC32.DLL starts a Win32 thread that periodically reads the selected serial port, making a copy of the incoming sentence. While one sentence is in the process of being read, the previously read sentence (which has been completely read) may be copied by the calling application program.

There are numerous sentence types, but the individual fields of any sentence can be read by the application. Seven sentence types (GPGGA, GPGLL, GPGSV, GPBOD, GPWPL, GPRMC, and GPVTG) have predefined field names and functions (such as mgcLatitude and mgcLongitude) that can retrieve specifically defined fields.

Note that the application program always has access to the most current sentence.

Calling **mgcClose** closes the serial port connection.

Typical program logic is as follows:

(1) Call **mgcAttach**, being sure to check the return code.
(2) Open the serial port with **mgcOpen**, being sure to check the return code.
(3) Choose sentence type to read (GPGGA, GPGLL, GPGSV, GPRMC, GPVTG, or all).
(4) Lock data buffer by calling **mgcLockData(1)**
(5) Read the fields that you want. For example, use **mgcGetData** to read raw sentence fields.
(6) Unlock data buffer by calling **mgcLockData(0)**
(7) Process the data just read. The data may be displayed, saved to disk, etc.
(8) Allow sufficient time for the next sentence to be read, which is close to 1 second.
(9) Loop back to step (4) above to read the next sentence or fall through to step (10)
(10) Close the connection by calling **mgcClose.**

For more details on program logic, refer to the example programs that are included with **MGC.**

6 NMEA 183 Sentence Structure

The **MarshallSoft GPS Component** (MGC) library is designed to read GPS data from the serial port according to the NMEA (National Marine Electronics Association) specification. For specification details, see

```
http://www.nmea.org
```

The NMEA 183 standard specifies 4800 baud, 8 bit data, and no stop bits.

MGC can read any GPS NMEA 183 sentence, extracting each field. In addition, it can read and decode the seven major sentences used for navigation: GPGGA, GPGLL, GPGSV, GPRMC, GPBOD, GPWPL and GPVTG.

The NMEA 0183 INTERFACE STANDARD (printed document) can be ordered from the National Marine Electronics Association (410-975-9425).

Version 6.3 of the "NMEA FAQ" contains details of the NMEA sentence structure and can be found at

```
http://vancouver-webpages.com/peter/nmeafaq.txt
```

7 Versions of the MarshallSoft GPS Component

The **MarshallSoft GPS Component (MGC)** library is available in three versions. All three versions have identical functionality.

7.1 Evaluation Version

The evaluation version can be differentiated from the other two versions by:

(1) The evaluation reminder screen is displayed at startup.

(2) The evaluation version may <u>not</u> be used for commercial purposes.

7.2 Academic Version

The academic version can be differentiated from the other two versions by:

(1) There is no evaluation reminder screen.

(2) The academic version may <u>not</u> be used for commercial purposes. It must be used for educational purposes only.

7.3 Professional Version

The professional version (developer license) can be differentiated from the other two versions by:

(1) There is no evaluation reminder screen.

(2) Source is available [see Section 1.4 above].

(3) The professional version may be used for commercial purposes.

The professional version may be distributed royalty free with your compiled application as specified by the software license. However, neither the License Key nor the source code to the MGC DLL (if purchased) may be distributed. See Section 1.4 "How to Purchase" for details on ordering and Section 9.2 for the license information.

8 Problems

(1) First, be sure you are passing the proper key code. Refer to Section 2.1 above.

(2) If you still get the registration reminder screen after registering the **MarshallSoft GPS Component** SDK, the problem is that Windows is finding the evaluation DLL before the registered DLL. Make sure you have deleted all copies of the **MGC** evaluation DLL in the Windows search path. The solution is to delete (or zip up) all evaluation versions of the MGC32.DLL.

(3) If you get "error -74" when calling **mgcAttach**, the problem is that the keycode passed to **mgcAttach** does not match the keycode in the DLL's. This is caused by using the evaluation keycode (value = 0) with the registered DLL.

(4) Are you sure that your serial port is working? Try using the serial port with an external modem or other serial device.

(5) Are you sure that your GPS unit is transmitting data? Run the RAW example program.

We recommend the following steps if you believe that you have discovered a bug in the library:

       (1) Create the smallest, simplest test program possible that demonstrates the problem.
       (2) Document your exact machine configuration and what error the test program demonstrates.
       (3) Email us the example source.

If the problem is an error in the library and can be solved with an easy work-around, we will publish the work-around. If the problem requires a modification to the library, we will make the change and make the modified library available to our customers without charge.

9 Legal Issues

9.1 License

MarshallSoft Computing, Inc. grants the registered user of MarshallSoft GPS Component (MGC) the right to use one copy of the MGC32.DLL on a single computer in the development of any software product. The user may not use the library on more than one computer at the same time. Licenses are per developer. One license must be purchased for each developer using the MarshallSoft GPS Component toolkit.

The "academic" registered DLL may not be distributed under any circumstances and may not be used for any commercial purpose.

The "professional" registered DLL may be distributed (without royalty) in object form only, as part of the user's compiled application. The registered DLL may NOT be distributed as part of any software development system (compiler or interpreter) without our express written permission. The source code for the library (MGC.C and WSC32MGC.C) is copyrighted by MarshallSoft Computing and may not be released in whole or in part.

9.2 Warranty

MARSHALLSOFT COMPUTING, INC. DISCLAIMS ALL WARRANTIES RELATING TO THIS SOFTWARE, WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND ALL SUCH WARRANTIES ARE EXPRESSLY AND SPECIFICALLY DISCLAIMED. NEITHER MARSHALLSOFT COMPUTING, INC. NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THIS SOFTWARE SHALL BE LIABLE FOR ANY INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH SOFTWARE EVEN IF MARSHALLSOFT COMPUTING, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR CLAIMS. IN NO EVENT SHALL MARSHALLSOFT COMPUTING, INC.'S LIABILITY FOR ANY SUCH DAMAGES EVER EXCEED THE PRICE PAID FOR THE LICENSE TO USE THE SOFTWARE, REGARDLESS OF THE FORM OF THE CLAIM. THE PERSON USING THE SOFTWARE BEARS ALL RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE.

Some states do not allow the exclusion of the limit of liability for consequential or incidental damages, so the above limitation may not apply to you.

This agreement shall be governed by the laws of the State of Alabama and shall inure to the benefit of MarshallSoft Computing, Inc. and any successors, administrators, heirs and assigns. Any action or proceeding brought by either party against the other arising out of or related to this agreement shall be brought only in a STATE or FEDERAL COURT of competent jurisdiction located in Madison County, Alabama. The parties hereby consent to in personam jurisdiction of said courts.

10 MGC Function Summary

Refer to the MGC Reference Manual (MGC_REF) for detailed information on the **MGC** functions. A one line summary of each function follows.

There are 30 functions in the **MarshallSoft GPS Component** library.

```
mgcAltitude        Returns last altitude value.
mgcAttach          Attaches MGC32.DLL.
mgcBearing         Returns bearing from one position to another.
mgcCallback        Registers callback function (for arriving data)
mgcClose           Closes MGC.
mgcDecodeDeg       Decodes integral degrees from coded integer.
mgcDecodeMin       Decodes integral minutes from coded integer.
mgcDecodeSec       Decodes integral seconds from coded integer.
mgcDecodeTho       Decodes integral thousandths of a minute from coded integer.
mgcEncodeDMS       Encodes (deg, min, sec) to coded integer.
mgcEncodeDMT       Encodes (deg, min, tho) to coded integer.
mgcErrorText       Returns error message text.
mgcGetData         Returns data field value.
mgcGetInteger      Returns named integer parameter.
mgcGetSentenceType Returns type of current "ready" sentence.
mgcGetString       Returns named string parameter.
mgcGreatCircle     Returns great circle distance.
mgcLatitude        Returns last latitude value.
mgcLockData        Locks (and unlocks) current data record.
mgcLongitude       Returns last longitude value.
mgcNewLatitude     Computes new latitude position.
mgcNewLongitude    Computes new longitude position.
mgcOpen            Open serial port for NMEA sentence input.
mgcRelease         Releases MGC32.DLL
mgcSeaLevel        Returns last sea level value.
mgcSetInteger      Sets named integer parameter.
mgcSleep           Sleeps specified time.
mgcStatus          Return TRUE (non 0) if data in being received.
mgcSysTime         Returns milliseconds since boot-up.
mgcTimestamp       Returns last timestamp value.
```

[END]