MarshallSoft GPS Component

Library for Visual Basic

Programmer's Manual

(MGC4VB)

Version 2.2

June 16, 2011

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

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

Voice: 1.256.881.4630
Web: www.marshallsoft.com

TABLE OF CONTENTS

1 Introduction

The **MarshallSoft GPS Component for Visual Basic (MGC4VB)** library is a software toolkit that allows software developers to quickly develop GPS applications in VB or VB.NET. The **MarshallSoft GPS Component (MGC)** is a library of functions providing direct and simple control of GPS data.

A straightforward interface provides the functionality to read and decode standard GPS (Global Positioning System) NMEA 183 sentences from the RS232 serial port as well as compute great circle distances and bearings.

This manual, the **MarshallSoft GPS Component Programmer's Manual,** provides information needed to compile programs in a Visual Basic or VB.NET environment.

The **MarshallSoft GPS Component for Visual Basic** library toolkit supports and has been tested with all Win32 and Win64 versions of Microsoft Visual Basic (VB 4/5/6) and Microsoft Visual Studio .NET (VB .NET). **MGC4VB** can also be used with Microsoft VBA (Visual Basic for Applications) such as EXCEL, ACCESS and Microsoft Office as well as with Power Builder.

MGC4VB runs under all versions of Windows (Windows 95, Windows 98, Windows ME, Windows 2000, Windows 2003, Windows NT, Windows XP, Windows 7, Vista and x64). The MarshallSoft GPS Component SDK DLLs (MGC32.DLL and MGC64.DLL) can also be used from any language (C/C++, Codegear Delphi, Visual FoxPro, COBOL, Xbase++, dBase, Microsoft Office, etc.) capable of calling the Windows API. Win32 and Win64 DLLs are provided.

Since the **MGC** toolkits use the same DLLs for all software languages, a developer can, for example, purchase **MGC4VB** and use the MGC DLLs also with **MGC4C** (C/C++ version).

MGC4VB includes multiple Visual Basic and VB. NET example programs with full source that demonstrate using MGC functions to read and decode GPS data. Our goal is to provide a robust GPS communications library that you and your customers can depend upon. Contact us if you have any questions.

When comparing the **MarshallSoft GPS Component** library toolkit against our competition, note that:

(1) **MGC4VB** is a standard Windows DLL (NOT an OCX or ActiveX control) and is much smaller than a comparable OCX or ActiveX control.
(2) A WIN32 DLL and Win64 DLL are included.
(3) **MGC4VB** does NOT depend on ActiveX or Microsoft Foundation Class (MFC) libraries or similar "support" libraries.
(4) **MGC4VB** is fully thread safe.
(5) The **MGC4VB** functions can be called from applications not capable of using controls.

We also provide a version of the **MarshallSoft GPS Component Library** for C/C++ (MGC4C). Each version of MGC uses the same DLL (MGC32.DLL or MGC64.DLL for 64-bit applications). However, the examples provided for each version are written for the specified computer language. We also have declaration files for other programming languages.

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

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

1.1 Features

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

- Supports both 32-bit and 64-bit Windows.
- Uses the Windows 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 GPGGA, GPRMC, GPGLL, GPGSA, GPVTG, GPBOD, GPWPL and GPGSV navigation sentences.
- Can graphically display latitude and longitude.
- Can compute distances and bearings.
- Provides ability to convert units.
- Works with Bluetooth serial.
- Work with USB ports that have a "USB to Serial Port" converter cable.

- **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.
- Can be used from GUI mode or console mode programs.
- Supports Windows 95/98/Me/NT/2000/2003/XP/Vista/Windows 7.
- **MGC** is implemented as a **standard** Windows DLL, which will work with all versions of Windows.
- Works with Microsoft Visual Basic (4.0,5.0,6.0) and Microsoft Visual Studio .NET (VB.NET)
- Works with Visual Basic for Applications (VBA) such as Excel, Microsoft Office and ACCESS.
- Will run on machines with or without .NET installed
- Does **not** depend on support libraries. Makes calls to core Windows API functions only.
- Can be used with any application program (in any language) capable of calling Windows API functions including C/C++, Visual Basic, PowerBASIC, Delphi, Visual FoxPro, Visual dBase, Xbase++, Fortran, COBOL, VBA, etc.
- The MGC license can be used with all supported computer programming languages.
- Can be purchased with (or without) ANSI C source code.
- Documentation online as well as in printable format.

A good selection of Visual Basic example programs with full source code is included. Refer to Section 6 for more details on each of the example programs. Refer to Section 4 for more details on each of the example programs.

[PROGRAM]	[DESCRIPTION]
MGCVER:	Program that displays the MGC4C version & build numbers.
HELLO:	Program similar to MGCVER but uses the MGC class fMGC.
GPRMC:	Program that decodes GPRMC sentences.
GPGGA:	Program that decodes GPGGA sentences.
GPGLL:	Program that decodes GPGLL sentences.
GPGSV:	Program that decodes GPGSV sentences.
GPVTG:	Program that decodes GPVTG sentences.
GPBOD:	Program that decodes GPBOD sentences.
GPWPL:	Program that decodes GPWPL sentences.
GPGSA:	Program that decodes GPGSA sentences.
GGA_RMC:	Program that decoded GPGGA & GPRMC sentences.
RAW:	Program that displays all sentences.
LATLON:	Program that graphically displays latitude & longitude.
COMPUTE:	Computes distances and bearings.
CONVERT:	Converts between (deg,min,min/1000), (deg,min,sec), etc.

1.2 Documentation Set

The complete set of documentation consists of three manuals. This is the first manual (MGC_4VB) in the set.

- MGC4VB Programmer's Manual (MGC_4VB.PDF)
- MGC User's Manual (MGC_USR.PDF)
- MGC Reference Manual (MGC_REF.PDF)

The **MGC_4VB Programmer's** language specific (Visual Basic) manual and provides information needed to install and compile example programs in a Visual Basic environment. Read this manual first.

The **MGC User's Manual** (MGC_USR) discusses GPS in general as well as language independent programming issues such as application notes, purchasing and licensing. Read this manual second.

The **MGC Reference Manual (MGC_REF)** contains details on each individual MGC function.

The online documentation can be accessed on the **MarshallSoft GPS Component for Visual Basic** product page at:

http://www.marshallsoft.com/mgc4vb.htm

1.3 Example Program

The following example code segment demonstrates the use of some of the **MarshallSoft GPS Component** library toolkit functions:

```
Code = mgcAttach(MGC_KEY_CODE)
If Code < 0 Then
  MsgBox "ERROR: Cannot attach. Check MGC_KEY_CODE."
  End
End If
' open port
Code = mgcOpen(ThePort)
If Code < 0 Then
  MsgBox "ERROR: Cannot open port."
  Exit Sub
End If
' port successfully opened, set sentence type (GPGGS) wanted
Code = mgcSetInteger(MGC_SET_SENTENCE_TYPE, MGC_SENTENCE_GPGGA)
While True
  ' lock data buffer
  Code = mgcLockData(1)
  ' get GPGGA sentence
  RefCount = mgcGetData(GPGGA_UTC_TIME, DataBuffer)
  If (RefCount < 0) And (RefCount <> MGC_NO_DATA) Then
    MsgBox "ERROR: " + Str$(RefCount)
    Exit Sub
  End If
  ' 1st time seen this data ?
  If RefCount = 1 Then
    ' get (coded) latitude (assume North in this example)
    iCoded = mgcLatitude()
    ' compute components
    LatDeg = mgcDecodeDeg(iCoded)
    LatMin = mgcDecodeMin(iCoded, LatDeg)
    LatSec = mgcDecodeSec(iCoded, LatDeg, LatMin)
    '
    ' display result as desired ...
    '
  End If

  ' unlock data buffer
  Code = mgcLockData(0)
Wend

Code = mgcClose()
Code = mgcRelease()
```

In the example program above, the serial port is opened and latitude is continuously read. The latitude components (degrees, minutes, and thousands of a minute) are then computed for display. See the MGC User's Manual (MGC_USR) for details on the various latitude/longitude component descriptions.

This example code segment was taken from the LATLON example program.

Refer to the MGC Reference Manual (MGC_REF) for individual function details.

1.4 Installation

(1) Before installation of MGC4VB, your Visual Basic compiler (any 32-bit or 64-bit version) should already be installed on your system and tested.

(2) Unzip MGC4VB22.ZIP (evaluation version) or MGCxxxx.ZIP (registered version where xxxx is your Customer ID) using any Windows unzip program.

(2) Run the installation program, SETUP.EXE, which will install all MGC4VB files including copying MGC32.DLL and MGC64.DLL to your Windows directory. Three sub-directories are created as follows:

```
DOCS – All documentation files
APPS – All example code
DLLS – All DLL’s
```

(3) You are ready to run! Note that no DLL registration is required.

1.5 Uninstalling

Uninstalling MGC4VB is very easy.

First, run UINSTAL.BAT, which will delete MGC32.DLL and MGC64.DLL from your Windows directory, typically C:\WINDOWS for Windows 95/98/Me/XP/2003/Vista/Win7 or C:\WINNT for Windows NT/2000.

Second, delete the MGC project directory created when installing MGC4VB.

1.6 Pricing

A developer license for the MarshallSoft GPS Component library can be purchased for $115 (or $195 with source code [Ansi C] to the library DLL). Purchasing details can be found in Section 1.4, "How to Purchase", of the MGC User’s Manual (MGC_USR).

Also see INVOICE.TXT or http://www.marshallsoft.com/order.htm. Registration includes one year of technical support and downloadable updates.

1.7 Updates

When a developer license is purchased for MGC4VB, the developer will be sent a new set of registered DLLs (MGC32.DLL and MGC64.DLL) plus a license file (MGCxxxx.LIC) that can be used to update the registered DLLs for a period of one year from purchase. Updates can be downloaded from

```
http://www.marshallsoft.com/update.htm
```

After one year, your license must be updated to be able to download updates. A developer license can be updated for:

- $30 if the update is ordered within one year of the original purchase (or previous update).
- $55 if the update is ordered between one and three years of the original purchase.
- $75 if the update is ordered after three years of the original purchase.

If source code was previously purchased, updates to the source code can be purchased for $40 along with the DLL update. A license update includes an additional year of technical support. Note that the registered DLL’s never expire.

2 Library Overview

2.1 Dynamic Link Libraries

The **MarshallSoft GPS Component Library (CSC)** is implemented as a dynamic link library (DLL) (Win32 and Win64). 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 the traditional static library that is bound to each and every application that uses it at link time.

An important advantage that DLL's have over other "popular" library formats such as VBX or OCX is that DLL's are callable by all Windows applications. Since DLL's are the building blocks of the Windows Operations System, they will not be replaced by a "newer technology".

2.2 Keycode

The MGC DLLs have a KeyCode encoded within them. Your keycode is a 9 or 10-digit decimal number (unless it is 0), and will be found in the file KEYCODE.BAS and KEYCODE.VB. The keycode for the evaluation version is 0. You will receive a new KeyCode after purchasing a license. During SETUP, the keycode is copied to the \APPS sub-directory.

If the error message (value -74) is received when calling **mgcAttach**, it means that the keycode in your application does not match the keycode in the DLL. After registering, it is best to remove the evaluation version of MGC32.DLL from the Windows search path.

2.3 Using the Library

The **MarshallSoft GPS Component Library for Visual Basic** has been tested on multiple computers running Windows 95/98/Me/XP/Vista/2003/Win7 and Windows NT/2000.

The **MGC4VB** library has been tested with all versions of Microsoft Visual Basic and Microsoft Visual Studio (VB.NET). **MGC4VB** can also be used with Visual Basic for Applications (VBA) such as Microsoft Office, Excel, and Access as well as with Power Builder.

The SETUP installation program will copy the Lib’s and DLL to the Windows directory. Refer to Section 1.4 "Installation".

2.4 MGC4VB Class

The **MGC** class "mgcClass" (mgcClass.cls) is a Visual Basic class wrapper for making calls to MGC32.DLL. The class name for each function is the same as the DLL function except the leading "mgc" is replaced by "f".

Those functions that return strings do so by use of the GetDataBuffer property. Instantiate **mgcClass** as any other class in Visual Basic:

```
Dim C As New mgcClass
```

Also refer to the **MarshallSoft GPS Component Reference Manual** (MGC_REF), the example project "HELLO" and the file mgcClass.cls. The use of **mgcClass** is limited to Visual Basic v5.0 and above since previous versions of Visual Basic do not support classes.

2.5 Dynamic Strings

The Visual Basic language uses a technique known as "garbage collection" to manage string space at runtime and may be called internally at any time by the Visual Basic runtime, asynchronous to what you may be doing in your code.

When passing a string buffer to a DLL function into which text will be copied, it is strongly recommended that the local string be allocated immediately before use. For example, a string buffer is passed to the user defined dllGetMessage function , which copies a text message into it. Note that `SPACE$(80)` is called immediately before dllGetMessage.

```
Dim Code As Integer
Dim Buffer As String * 80
' allocate buffer just before call to dllGetMessage
Buffer = SPACE$(80)
' copy message into 'Buffer'
Code = dllGetMessage(Buffer, 80)
' message text is now in 'Buffer'
```

This technique is not necessary for passing a string <u>to</u> a DLL function, only when passing a buffer to a DLL into which data is to be placed by the DLL function.

2.6 Adding MGC4VB to Your Project

Copy MGC32.BAS (if running VB 4/5/6), or MGC32.VB (if running Visual Studio/VB.Net) into the same directory (folder) as the application program to which you want to add MGC code. You will find these files in the APPS sub-directory (folder) created when you ran SETUP, usually C:\MGC4VB\APPS.

2.6.1 Adding MGC4VB to your VB_4.0 .0, 5.0, or 6.0 Project

Open your existing project with "File", "Open Project". Then choose "Insert", "Module", then add MGC32.BAS and KEYCODE.BAS to your project. If prompted to add "DAO 2.50 Object Library", choose "no".

MGC functions can now be called from your VB program.

2.6.2 Adding MGC4VB to your VB.Net Project

Open your existing project with "File", "Open Project". Then choose "Project", "Add Module", then add MGC32.VB and KEYCODE.VB to your project. MGC functions can now be called from your VB.NET program.

Substitute MGC32 for MGC64 for 64-bit applications.

2.7 Error Display

The error message text associated with MGC error codes can be displayed by calling **mgcErrorText**.

```
Function GetErrorText(ByVal ErrorCode As Long) As String
Dim Code As Long
Dim Text As String * 128
Code = mgcErrorText(ErrorCode, Text)
If Code > 0 Then
  GetErrorText = Text
Else
  GetErrorText = "Error code " + LTrim$(Str$(ErrorCode))
End If
End Function
```

3. Compiler Issues

The **Marshall GPS Component for Visual Basic** component library supports and has been tested with all versions of Microsoft Visual Basic (VB 3, VB 4, VB 5, VB 6 and Microsoft Visual Studio .through Visual Studio 2010.

3.1 Visual Basic Makefiles

Project files for 32-bit Visual Basic end with ".VBP". Projects for Visual Studio (VB.NET) end with ".vbproj".

3.2 Compiling Programs

Choose "File", then "Open Project" from the main Visual Basic menu.

When saving the example programs in VB 5 or VB 6 format, answer "no" if asked to add the "Microsoft DAO v2.5 library".

Compile and run MGCVER as the first example. MGCVER should display the version number from MGC32.DLL (or MGC64.DLL for 64-bit applications).

3.3 VB.Net and Visual Studio

There are a few differences between VB 4/5/6 and VB.NET / Visual Studio that affect writing programs that use MGC.

(1) Variables that are declared "As Long" in VB4/5/6 are declared "As Integer" in VB.Net.

(2) Fixed length strings are not supported in VB.Net. When calling any MGC function that can return a string (**mgcGetString** for example), memory for the string variable must be allocated first. For example:

```
Dim Buffer As String
Buffer = Space(80)
Code = mgcGetString(0, MGC_GET_REGISTRATION, Buffer, 80)
```

(3) Some Visual Basic functions must be fully qualified. For example, instead of LEFT, use `Microsoft.VisualBasic.Left`

(4) The module MGC32.VB , or MGC64.VB for 64-bit programs, must be included in all Visual Studio / VB.NET programs.

3.4 Visual Basic for Applications (VBA)

The MarshallSoft GPS library can be used with Microsoft VBA applications such as EXCEL, ACCESS, and Microsoft Office.

Start EXCEL (or other 32-bit Office VBA program such as WORD or ACCESS), then enter design mode. Enable the "Controls Toolbox", choose "Tools" on the menu bar, then "Customize", and then check "Control Toolbox". From the control toolbox, choose and position a "Command Button". This will create code that looks like

```
     Private Sub CommandButton1_Click()

     End Sub
```

Replace the generated code with MODULE32.BAS. The easiest way to do this is to paste from the clipboard.

Exit design mode and then press the command button to start the VBA example program.

3.5 Power Builder

MGC can also be used with 32 bit Power Builder applications. See PBUILDER.TXT in the \APPS subdirectory for more information.

```
     MGC32.PBI : Power Builder declaration file.
```

4 Example Programs

Several example programs are included in MGC4VB. There is also an example program (HELLO), which uses the VB class "mgcClass", but requires VB 5 or VB 6 (which support classes).

Example programs are located in the \APPS sub-directory created by SETUP.

4.1 MGCVER

The first example program is the program MGCVER (MGC Version) that displays the MGC library version number and registration string. Its purpose is display the MGC version, build, and registration string as well as to verify that MGC32.DLL is being found and loaded by Windows.

The project files are:

```
MGCVER.VBP    : for (32-bit) VB 4.0 and above.
MGCVER.VBPROJ : for Visual Studio / VB.Net
```

4.2 HELLO

The HELLO example program is similar to MGCVER except that it uses the MGC class **mgcClass.cls**. See mgcClass.cls. Visual Basic 5 or above is required for programs using VB classes.

4.3 COMPUTE

COMPUTE is a Visual Basic program that computes great circle distances and bearings from a pair of latitude/longitude values.

The project files are:

```
COMPUTE.VBP    : for (32-bit) VB 4.0 and above.
COMPUTE.VBPROJ : for Visual Studio / VB.Net.
COMPUTE(VS2008).VBPROJ    : for 32-bit Visual Studio 2008.
COMPUTE(VS2008)X64.VBPROJ : for 64-bit Visual Studio 2008.
COMPUTE(VS2010).VBPROJ    : for 32-bit Visual Studio 2010.
COMPUTE(VS2010)X64.VBPROJ : for 64-bit Visual Studio 2010.
```

4.4 CONVERT

CONVERT is a Visual Basic program that converts latitude (or longitude) between (deg, min, min/1000), (deg, min, sec), and single integer encoding.

The project files are:

```
CONVERT.VBP    : for (32-bit) VB 4.0 and above.
```

4.5 LATLON

LATLON is a VB example program that continuously displays latitude and longitude.

The project files are:

```
LATLON.VBP    : for (32-bit) VB 4.0 and above.
LATLON.VBPROJ : for Visual Studio / VB.Net.
LATLON(VS2008).VBPROJ    : for 32-bit Visual Studio 2008.
LATLON(VS2008)X64.VBPROJ : for 64-bit Visual Studio 2008.
LATLON(VS2010).VBPROJ    : for 32-bit Visual Studio 2010.
LATLON(VS2010)X64.VBPROJ : for 64-bit Visual Studio 2010.
```

4.6 GPGGA

GPGGA is a VB program that displays all NMEA GPGGA (Global Positioning System Fix Data) sentences.

The project files are:

```
GPGGA.VBP    : for (32-bit) VB 4.0 and above.
GPGGA.VBPROJ : for Visual Studio / VB.Net.
GPGGA(VS2008).VBPROJ    : for 32-bit Visual Studio 2008.
GPGGA(VS2008)X64.VBPROJ : for 64-bit Visual Studio 2008.
GPGGA(VS2010).VBPROJ    : for 32-bit Visual Studio 2010.
GPGGA(VS2010)X64.VBPROJ : for 64-bit Visual Studio 2010.
```

4.7 GPGLL

GPGLL is a VB program that displays all NMEA GPGLL (Geographic Position, Latitude / Longitude) sentences.

The project files are:

```
GPGLL.VBP    : for (32-bit) VB 4.0 and above.
GPGLL.VBPROJ : for Visual Studio / VB.Net.
GPGLL(VS2008).VBPROJ    : for 32-bit Visual Studio 2008.
GPGLL(VS2008)X64.VBPROJ : for 64-bit Visual Studio 2008.
GPGLL(VS2010).VBPROJ    : for 32-bit Visual Studio 2010.
GPGLL(VS2010)X64.VBPROJ : for 64-bit Visual Studio 2010.
```

4.8 GPGSV

GPGSV is a VB program that displays all NMEA GPGSV (GPS Satellites in View) sentences.

The project files are:

```
GPGSV.VBP    : for (32-bit) VB 4.0 and above.
```

4.9 GPRMC

GPRMC is a VB program that displays all NMEA GPRMC (Recommended Minimum Specific GPS/Transit Data) sentences.

The project files are:

```
     GPRMC.VBP    : for (32-bit) VB 4.0 and above.
```

4.10 GPVTG

GPVTG is a VB program that displays all NMEA GPVTG (Track Made Good and Ground Speed) sentences.

The project files are:

```
     GPVTG.VBP    : for (32-bit) VB 4.0 and above.
```

4.11 RAW

RAW is a VB program that displays all NMEA sentences.

The project files are:

```
     RAW.VBP    : for (32-bit) VB 4.0 and above.
     RAW.VBPROJ : for Visual Studio / VB.Net.
     RAW(VS2008).VBPROJ    : for 32-bit Visual Studio 2008.
     RAW(VS2008)X64.VBPROJ : for 64-bit Visual Studio 2008.
     RAW(VS2010).VBPROJ    : for 32-bit Visual Studio 2010.
     RAW(VS2010)X64.VBPROJ : for 64-bit Visual Studio 2010.
```

4.12 GPBOD

GPBOD is a VB program that displays all NMEA GPBOD (Bearing, Origin to Destination) sentences.

The project files are:

```
     GPBOD.VBP    : for (32-bit) VB 4.0 and above.
```

4.13 GPWPL

GPWPL is a VB program that displays all NMEA GPWPL (Waypoint Location) sentences.

The project files are:

```
     GPWPL.VBP    : for (32-bit) VB 4.0 and above.
```

4.14 GPGSA

GPWPL is a VB program that displays all NMEA GPWPL (GPS DOP and active satellites) sentences.

The project files are:

```
     GPGSA.VBP    : for (32-bit) VB 4.0 and above.
```

5 Revision History

The MarshallSoft GPS Component DLL (MGC32.DLL) is written in ANSI C. All language versions of MGC (C/C++, Delphi, Visual Basic, PowerBASIC, etc) use the same identical DLL.

Version 1.2: February 7, 2003.

- The initial release of the Visual Basic version MGC4VB.

Version 1.3: August 9, 2004

- Increased maximum number of fields from 19 to 20.
- Receive buffer size increased from 256 to 1024.
- Overflow wraps to empty.
- Added MGC_SET_SLEEP to mgcSetInteger (sets work buffer sleep time).
- Added mgcStatus() function to return GPS data status.

Version 1.4 March 3, 2005

- Added $GPVTG sentence type.
- Interval diagnostics added.

Version 2.0: July 26, 2007

The "Coded Integer" angles returned by MGC functions are now returned in units of hundred-thousandths of a degree rather than thousandths of a minute.

- mgcBearing returns the bearing in units of hundred-thousandths of a degree (100000 units = 1 deg).
- Two sentence types can be specified rather than just one. See GGA_RMC.C
- mgcGetSentenceType added that returns the current sentence type (defined in mgc.bas)
- mgcNewLatitude added that returns a new latitude at a specified distance.
- mgcNewLongitude added that returns a new longitude at a specified distance.

Version 2.1: March 2, 2009

- mgcSetInteger (MGC_SET_BAUDRATE, baud-rate) changes baud rate immediately.
- Added GPBOD sentence type.
- Added GPWPL sentence type.
- Number of ports increased to 256 (COM1 through COM256).
- Added documentation in Adobe PDF format.

Version 2.2: June 16, 2011

- Added support for Win-64.
- Added GPGSA sentence type.
- Fixed problem with leaking thread handle.

[END]