



Multi Port GPS Users Guide and Reference

for Windows XP, Windows 2000, and Windows NT
Constellation Data Systems, Inc.

www.VirtualPeripherals.com

Copyright © 2001-2005 Constellation Data Systems, Inc ("CDS"). All rights reserved. Consult your software license agreement. Brand and product names are trademarks of their respective holders. Portions of this manual are © Microsoft Corporation, and are used by permission of the MSDN.

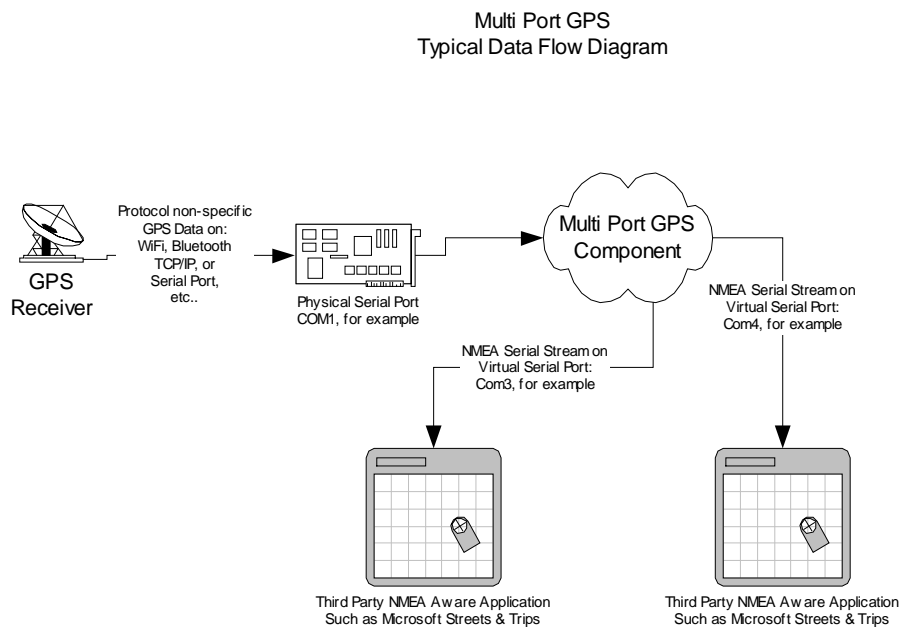
Table of Contents

1. Introduction	3
1.1 Overview	4
1.2 Capabilities	4
2. Installation Instructions	5
3. Multi Port GPS Users Guide	6
3.1 Command Line Parameters of Multi Port GPS.....	6
3.2 Help Screen	11
3.3 Demonstration Using Multi Port GPS.....	11
4. Supplemental VSP Utilities	14
4.1 Add Port Utility	14
4.1.1 Demonstration of <i>Add Port</i>	14
4.2 Delete Port Utility	15
4.2.1 Demonstration of <i>Delete Port</i>	15
4.3 Enum Ports Utility	16
4.3.1 Demonstration of <i>Enum Ports</i>	16
5. Multi Port GPS Custom Programming.....	17
5.1 Multi Port GPS Sample Reference Design (C/C++).....	17
6. Notices.....	18
7. Index of Acronyms and Abbreviations.....	19

1. Introduction

Constellation Data System's Multi Port GPS component is a powerful pre-built Virtual Serial Port (VSP) application. Many users will be able to use this utility in stand-alone form. Other users, with more complex, custom requirements may wish to consider developing custom applications using the VSP Software Development Kit (VSP SDK).

Consider the following data flow diagram which illustrates the basic data flow of a system using Multi Port GPS as a component.



1.1 Overview

Multi Port GPS demonstrates the techniques of taking GPS data, i.e. NMEA protocol sentences, from a single GPS receiver device and replicating that NMEA data onto multiple Virtual Serial Ports. Multi Port GPS also re-structures corrupt NMEA sentences and restores the necessary timing relationships. An engineer may wish to use this sample as a starting point for a GPS data splitter. The following examples show two Virtual Serial Ports, however the utility is designed to work with an arbitrary number of virtual serial ports.

1.2 Capabilities

- Splitting NMEA data from a single source onto multiple virtual serial ports.
- NMEA protocol compliant.
- Hardware-less serial port replication.
- Easy capture of data from a serial port data.
- High speed data transfers / transmissions.
- Multiplexing multiple data sources on a single serial port.
- Serial device simulation.
- Serial port data redirection and device simulation.
- Serial protocol development and implementation.

2. Installation Instructions

In overview, the installation consists of installing Constellation Data System's VSP Framework Core product, which can be downloaded from www.VirtualPeripherals.com/VspDownload.htm.

The VSP Core Users Guide and Reference Manual, which describes the VSP Core's installation and operation, can also be downloaded from www.VirtualPeripherals.com/VspDownload.htm. Be sure to review this document prior to installation and use.

After the core product is installed,

1. The setup executable is named using the following nomenclature: "VirtualSerialXXX.exe", where XXX is a short version description embedded in the file name. For example, version 2.27 of the Core VSP, XXX would be 227, and the entire module would be named "VirtualSerial227.exe". Run this executable.
2. After running the "VirtualSerialXXX.exe" executable, you will be prompted for the "Unzip To Folder". The folder "c:\VirtualSerial" is recommended. You will then be prompted to run "setup.exe". You should run that component.
3. After running "setup.exe" an Install Shield script will walk you through a number of rather straight-forward dialog boxes. For more detailed installation instructions, please refer to the *VSP Core Users Guide*.
4. After Installation is complete, you should create 2 serial ports (see Add Port Utility).
5. After placing the Multi Port GPS utility executable in the VSP installation directory (c:\VirtualSerial\exe), you may then run Multi Port GPS utility from a Windows Command Prompt.

3. Multi Port GPS Users Guide

Multi Port GPS replicates GPS data received on a physical serial port onto multiple Virtual Serial Ports.

Important:	<p>It is important that the source of the NMEA data is correctly identified (the first parameter) to Multi Port GPS (see below). The Virtual Serial Ports which act as the destinations (where an application such as Microsoft's Streets and Trips would connect), should be identified as subsequent parameters to Multi Port GPS (see below).</p> <p>Switching the source and destination parameters to Multi Port GPS will cause improper behavior.</p>
Note:	<p>Many serial port aware applications, such as Microsoft's Streets and Trips will only function on ports named COM1, COM2, COM3, or COM4.</p>

3.1 Command Line Parameters of Multi Port GPS

Consider the following syntax mapping:

Syntax	<p>MultiPortGps Source Destination1 [Destination2 ...] [switches]</p> <p>Source → Serial port which receives the NMEA stream. Destination1, Destination2 . . . → Virtual Serial Port(s) where applications such as Microsoft's Streets and Trips would connect.</p> <p>[switches] – Optional settings of the GPS COM port unless otherwise noted ('Virtual' nomenclature). All VALUE's are in decimal.</p> <pre> /Baud:VALUE /Parity:{EVEN MARK NO ODD SPACE} /RxParity:{ENABLE DISABLE} /StopBits:{ONE TWO ONEANDHALF} /OutCtsFlow:{ENABLE DISABLE} /DtrControl:{DISABLE ENABLE HANDSHAKE} /DsrSensitive:{ENABLE DISABLE} /TxContinueOnXoff:{ENABLE DISABLE} /OutX:{ENABLE DISABLE} </pre>
--------	--

	<p>/InX:{ENABLE DISABLE} /XonLim:VALUE /XoffLim:VALUE /XoffChar:VALUE /XonChar:VALUE /NullDiscard{ENABLE DISABLE} /RtsControl:{DISABLE ENABLE HANDSHAKE TOGGLE} /ErrorReplaceChar:VALUE /ByteSize:VALUE /ReadIntervalTimeout:VALUE /MonitorHex /MonitorAscii /DelayWriteFileByteToByte:VALUE /DelayWriteFileConstant:VALUE /DelayReadFileByteToByte:VALUE /DelayReadFileConstant:VALUE</p>
<p>Parameters</p>	<p>Source – The GPS communications port to use. This device must be an industry standard GPS communications device running the standard Windows serial port device driver. Data received at the indicated physical GPS port is replicated (written) to the all Virtual Serial Port(s) specified.</p> <p>Destination1, Destination2 ... – One or more Virtual Serial Ports. Data received at these ports are written to the GPS COM port.</p>
<p>Switches</p>	<p>/Baud:VALUE – The valid decimal data rate VALUE in bits per second (bps) for the physical port. Examples of valid data rates are 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, and so on.</p> <p>/Parity:{EVEN MARK NO ODD SPACE} - Parity scheme to be used in transmitting data and receiving data on the GPS COM port.</p> <p>/RxParity:{ENABLE DISABLE} – Indicates whether parity checking is enabled for data received on the GPS COM port. If enabled, parity checking is performed.</p> <p>/StopBits:{ONE TWO ONEANDHALF} - Number of stop bits to be used for data transmitted and received on the GPS COM port.</p> <p>/OutCtsFlow:{ENABLE DISABLE} - Indicates whether the GPS COM port's CTS (clear-to-send) signal is monitored for output flow control. If enabled and CTS is turned off, output is suspended until CTS is sent again.</p> <p>/DtrControl:{DISABLE ENABLE HANDSHAKE} – DTR (data-terminal-ready) flow control of the GPS COM port. DISABLE causes the DTR line to disable (de-assert)</p>

<p>Switches (continued)</p>	<p>when the GPS device is opened and leaves it disabled. ENABLE causes the DTR line to assert when the GPS device is opened and leaves it asserted. HANDSHAKE allows DTR handshaking on the GPS device, if handshaking is enabled</p> <p>/DsrSensitive:{ENABLE DISABLE} - Indicates whether the GPS COM port's communications driver is sensitive to the state of the DSR signal. If enabled, the GPS' device driver ignores any bytes received, unless the DSR modem input line is high (asserted).</p> <p>/TxContinueOnXoff:{ENABLE DISABLE} - Indicates whether transmission stops when the input buffer is full and the GPS' device driver has transmitted the XoffChar character. If enabled, transmission continues after the input buffer has come within XoffLim bytes of being full and the driver has transmitted the XoffChar character to stop receiving bytes. If not enabled, transmission does not continue until the input buffer is within XonLim bytes of being empty and the driver has transmitted the XonChar character to resume reception.</p> <p>/OutX:{ENABLE DISABLE} Indicates whether XON/XOFF flow control is used during transmission on the GPS COM port. If enabled, transmission stops when the XoffChar character is received and starts again when the XonChar character is received</p> <p>/InX:{ENABLE DISABLE} - Indicates whether XON/XOFF flow control is used during reception on the GPS COM port. If enabled, the XoffChar character is sent when the input buffer comes within XoffLim bytes of being full, and the XonChar character is sent when the input buffer comes within XonLim bytes of being empty.</p> <p>/XonLim:VALUE – On the GPS COM port, the minimum number of bytes allowed in the input buffer before flow control is activated to inhibit the sender. Note that the sender may transmit characters after the flow control signal has been activated, so this value should never be zero. This assumes that XON/XOFF, RTS, or DTR input flow control is specified using /InX, /RtsControl, or /DtrControl.</p> <p>/XoffLim:VALUE – On the GPS COM port, the maximum number of bytes allowed in the input buffer before flow control is activated to allow transmission by the sender. This assumes that XON/XOFF, RTS, or DTR input flow control is specified in /InX, /RtsControl, or /DtrControl. The maximum number of bytes allowed is calculated by subtracting this value from the size, in bytes, of a buffer specific to the GPS' device driver.</p> <p>/XonChar:VALUE - Value of the XON character for both transmission and reception on the GPS COM port.</p> <p>/XoffChar:VALUE - Value of the XOFF character for both transmission and reception on the GPS COM port.</p>
--	--

<p>Switches (continued)</p>	<p>/NullDiscard{ENABLE DISABLE} - Indicates whether null bytes are discarded. If enabled then null bytes are discarded when received on the GPS COM port.</p> <p>/RtsControl:{DISABLE ENABLE HANDSHAKE TOGGLE} - RTS (request-to-send) flow control of the GPS COM port. DISABLE causes the RTS line to be de-asserted (disabled) when the GPS device is opened and leaves it de-asserted (disabled). ENABLE asserts (enables) the RTS line when the GPS device is opened and leaves it asserted (on). HANDSHAKE allows RTS handshaking. The GPS COM port's driver asserts (raises) the RTS line when the "type-ahead" (input) buffer is less than one-half full and de-asserts (lowers) the RTS line when the buffer is more than three-quarters full. TOGGLE, which is valid in Windows NT/2000/XP, specifies that the RTS line will be asserted (high) if bytes are available for transmission. After all buffered bytes have been sent, the RTS line will be de-asserted (low).</p> <p>/ErrorReplace:{ENABLE DISABLE} – Indicates whether bytes received on the GPS COM port with parity errors are replaced with the character specified by the /ErrorReplaceChar value. If enabled, and the /Parity is enabled, then replacement occurs.</p> <p>/ErrorReplaceChar:VALUE – Value of the character used to replace bytes received on the GPS COM port, with a parity error.</p> <p>/ByteSize:VALUE - Number of bits in the bytes transmitted and received on the GPS COM port.</p> <p>/ReadIntervalTimeout:VALUE - Maximum time, in milliseconds, allowed to elapse between the arrival of two characters on the GPS COM port communications line. During a WIN32 ReadFile operation, the time period begins when the first character is received. If the interval between the arrivals of any two characters exceeds this amount, the WIN32 ReadFile operation is completed and any buffered data is returned. A value of zero indicates that interval time-outs are not used.</p> <p>/MonitorHex – prints a data trace on the console in Hex.</p> <p>/MonitorAscii – prints a data trace on the console in ASCII.</p> <p>/DelayWriteFileByteToByte:VALUE - Specifies the amount of time, in milliseconds, to delay between bytes when writing to the Virtual Serial Port(s)</p> <p>/DelayWriteFileConstant:VALUE - Specifies a constant amount of time, in milliseconds, to delay between bytes when writing to the Virtual Serial Port(s)</p> <p>/DelayReadFileByteToByte:VALUE - Specifies the amount of time, in milliseconds, to delay between bytes when reading to the Virtual Serial Port(s)</p> <p>/DelayReadFileConstant:VALUE - Specifies a constant amount of time, in</p>
--	---

	<p>milliseconds, to delay between bytes when reading to the Virtual Serial Port(s)</p> <p>***** IMPORTANT *****</p> <p>NOTE: If file timing is specified, it applies to each and every Virtual Serial Port specified on the command line.</p>
Runtime	<p>'Q' then 'ENTER', on the keyboard causes the utility to exit.</p>

3.2 Help Screen

Running “MutliPortGps.exe” with no parameters will cause the following dialog box to appear.

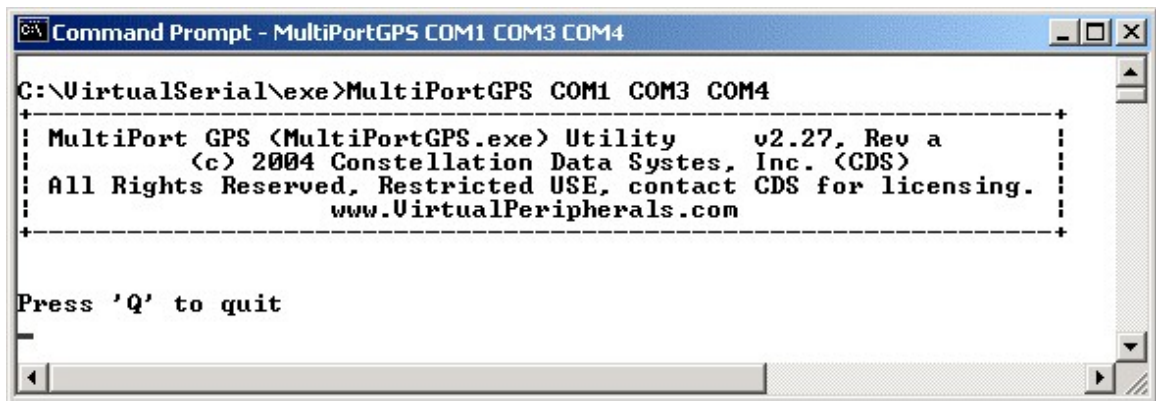


3.3 Demonstration Using Multi Port GPS

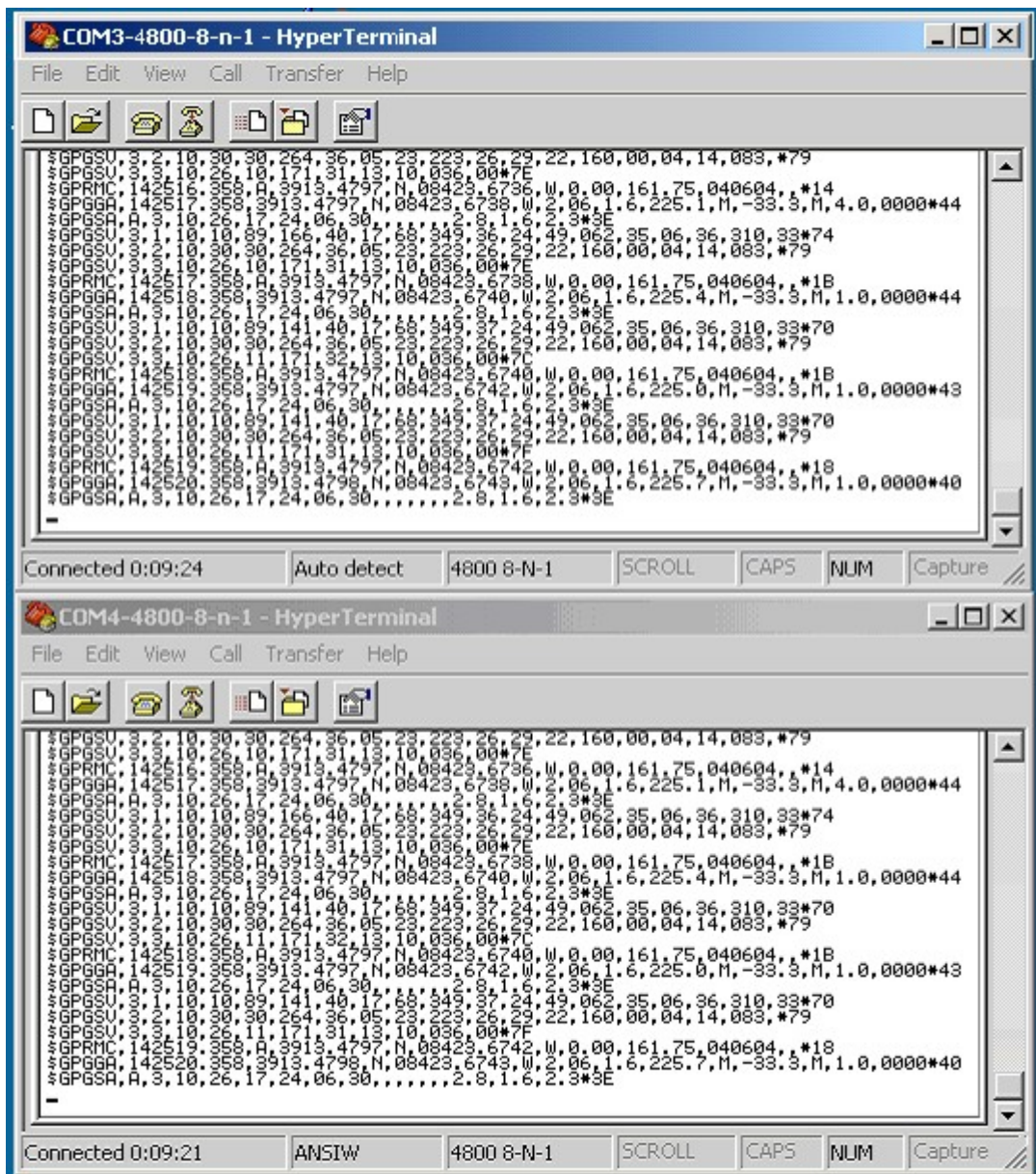
In the following example, the Multi Port GPS utility has been used to connect the Transmit (TX) data of the serial port “COM1” (GPS device) with the Receive (RX) data of Virtual Serial Ports “COM3” and “COM4”. Additionally, the Transmit (TX) data of Virtual Serial Ports “COM3” and “COM4” has been connected with the Receive (RX) data of serial port “COM1”(GPS device).

Important:	<p>It is important that the source of the NMEA data is correctly identified (the first parameter) to Multi Port GPS (see below). The Virtual Serial Ports which act as the destinations (where an application such as Microsoft's Streets and Trips would connect), should be identified as subsequent parameters to Multi Port GPS (see below).</p> <p>Switching the source and destination parameters to Multi Port GPS will cause improper behavior.</p>
------------	---

Consider the following execution which replicates the NMEA data written to COM1 (GPS device) onto both COM3 and COM4 (Virtual Serial Ports). Once executed, a third party application, such as Microsoft's Streets and Trips may be connected to COM3 or COM4



```
Command Prompt - MultiPortGPS COM1 COM3 COM4
C:\VirtualSerial\exe>MultiPortGPS COM1 COM3 COM4
+-----+
| MultiPort GPS (MultiPortGPS.exe) Utility      v2.27, Rev a |
|          (c) 2004 Constellation Data Syses, Inc. (CDS)    |
| All Rights Reserved, Restricted USE, contact CDS for licensing. |
|          www.VirtualPeripherals.com                |
+-----+
Press 'Q' to quit
```



The Multi Port GPS utility continues to operate until 'Q' and ENTER is typed on the console. This causes a QUIT operation and Multi Port GPS will gracefully shutdown and say "...Goodbye...".

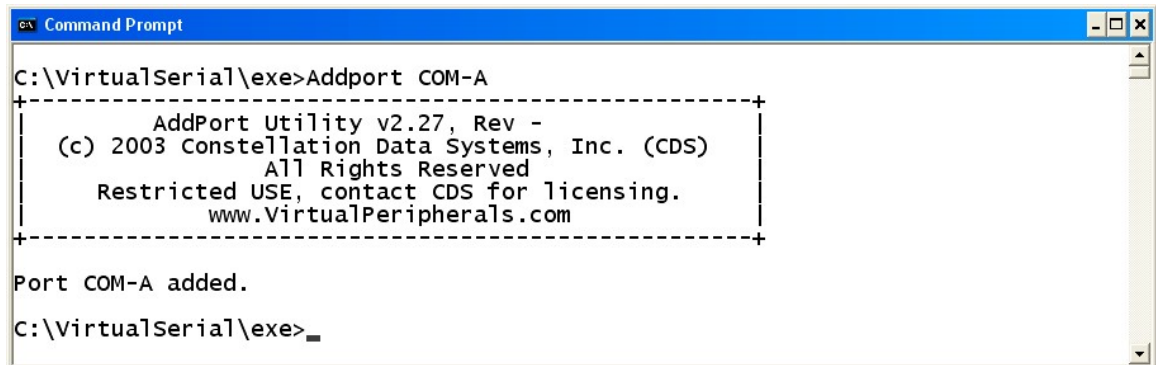
4. Supplemental VSP Utilities

4.1 Add Port Utility

The *Add Port* (“addport.exe”) utility creates a Virtual Serial Port. The VSP name will be the same name that applications, such as HyperTerminal use to identify the VSP device. If a port with the selected name already exists, a message will appear which explains why the Virtual Serial Port could not be created.

4.1.1 Demonstration of *Add Port*

In the following example, the *Add Port* utility has been used to create/add a Virtual Serial Port “PORTA”.

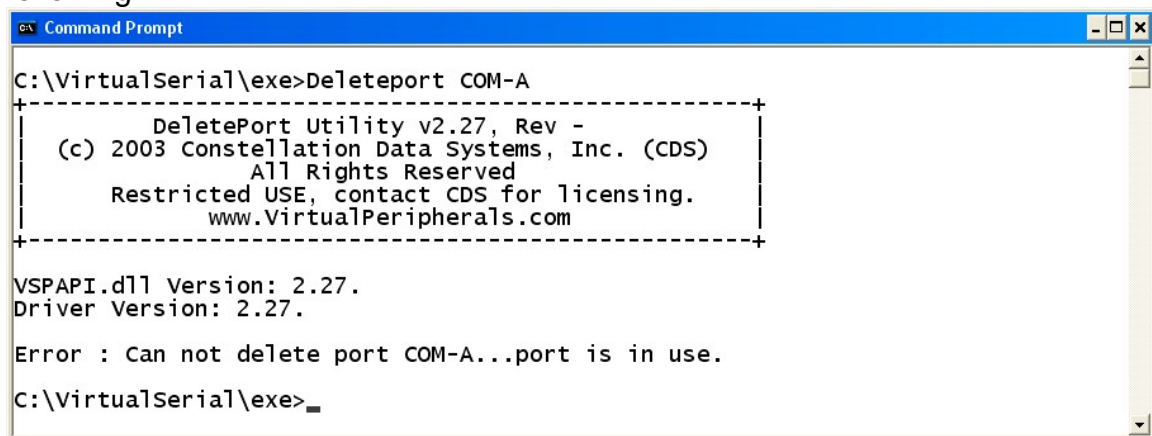


```
Command Prompt
C:\VirtualSerial\exe>Addport COM-A
-----+-----
      AddPort Utility v2.27, Rev -
      (c) 2003 Constellation Data Systems, Inc. (CDS)
      All Rights Reserved
      Restricted USE, contact CDS for licensing.
      www.VirtualPeripherals.com
-----+-----
Port COM-A added.
C:\VirtualSerial\exe>
```

4.2 Delete Port Utility

The *Delete Port* (“deleteport.exe”) utility deletes a Virtual Serial Port. The VSP name will be the same name that applications, such as HyperTerminal, use to identify the VSP device.

If a port with the selected name is currently in use, a message will appear which explains why the Virtual Serial Port could not be deleted. Should a virtual serial port device be locked by another component then the utility may fail. Should the device be locked, you will observe output similar to the following:

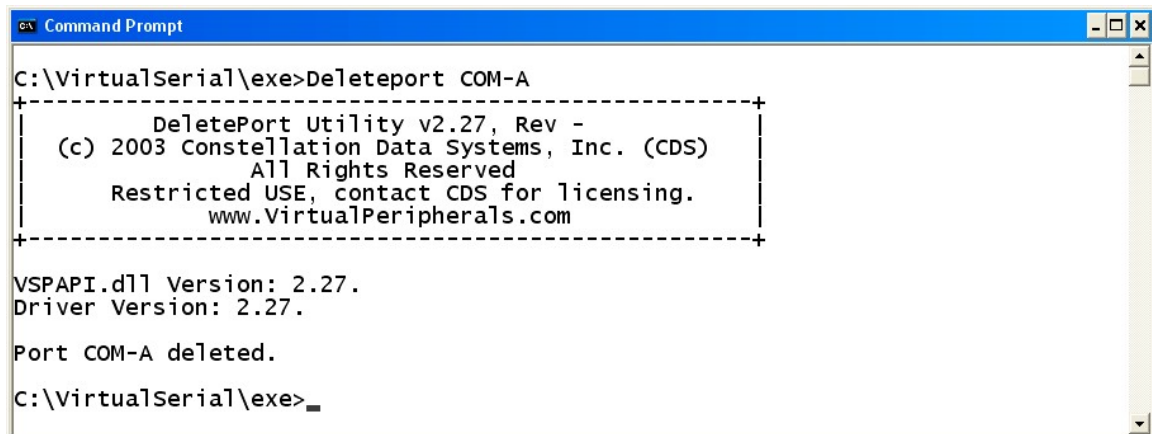


```
Command Prompt
C:\VirtualSerial\exe>Deleteport COM-A
+-----+
| DeletePort Utility v2.27, Rev - |
| (c) 2003 Constellation Data Systems, Inc. (CDS) |
| All Rights Reserved |
| Restricted USE, contact CDS for licensing. |
| www.VirtualPeripherals.com |
+-----+
VSPAPI.dll Version: 2.27.
Driver Version: 2.27.
Error : Can not delete port COM-A...port is in use.
C:\VirtualSerial\exe>_
```

To clear this condition, free the device. In this case, simply close the application (such as HyperTerminal), which is connected to the Virtual Serial Port.

4.2.1 Demonstration of *Delete Port*

In the following example, the *Delete Port* utility has been used to delete a Virtual Serial Port “PORTA”.



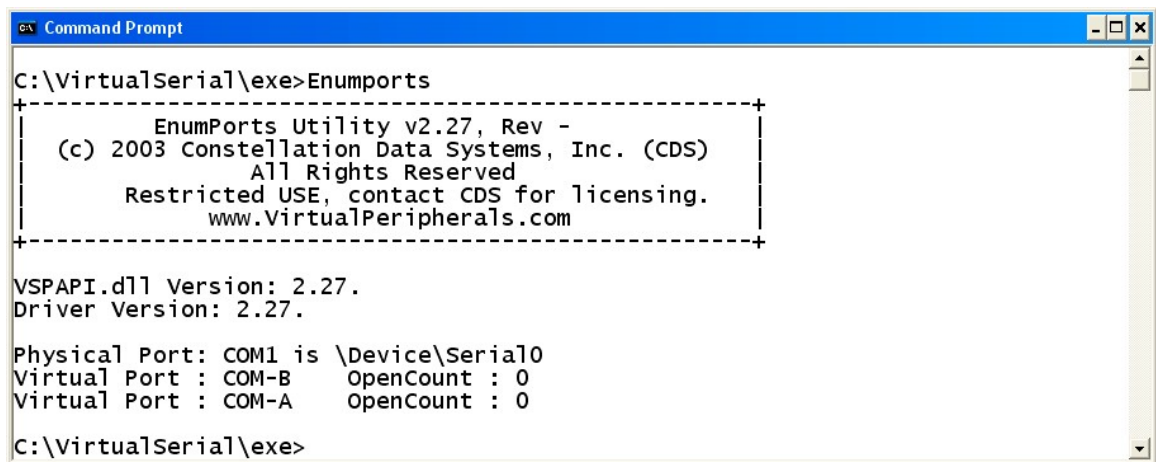
```
Command Prompt
C:\VirtualSerial\exe>Deleteport COM-A
+-----+
| DeletePort Utility v2.27, Rev - |
| (c) 2003 Constellation Data Systems, Inc. (CDS) |
| All Rights Reserved |
| Restricted USE, contact CDS for licensing. |
| www.VirtualPeripherals.com |
+-----+
VSPAPI.dll Version: 2.27.
Driver Version: 2.27.
Port COM-A deleted.
C:\VirtualSerial\exe>_
```

4.3 Enum Ports Utility

The *Enum Ports* (“enumports.exe”) utility enumerates all Virtual and Physical Serial Ports on a system.

4.3.1 Demonstration of *Enum Ports*

In the following example, the *Enum Ports* utility has been used to enumerate all Virtual and Physical Serial Ports.



```
C:\VirtualSerial\exe>Enumports

+-----+
|          EnumPorts Utility v2.27, Rev -          |
| (c) 2003 Constellation Data Systems, Inc. (CDS)  |
|          All Rights Reserved                    |
|          Restricted USE, contact CDS for licensing. |
|          www.VirtualPeripherals.com              |
+-----+

VSPAPI.dll Version: 2.27.
Driver Version: 2.27.

Physical Port: COM1 is \Device\Serial0
Virtual Port : COM-B   OpenCount : 0
Virtual Port : COM-A   OpenCount : 0

C:\VirtualSerial\exe>
```

5. Multi Port GPS Custom Programming

To start developing customized VSP applications, download and refer to the VSP SDK Programmers Guide and Reference from <http://www.virtualperipherals.com/VspDownloads/VSPSoftwareDevelopmentKit.pdf>.

5.1 Multi Port GPS Sample Reference Design (C/C++)

The source code for Multi Port GPS may be licensed from Constellation Data Systems (CDS). Please contact CDS sales for more information.

6. Notices

Use of this software, information, or technology in a system, or as a component of a system, which can through action or inaction, cause damage to life, limb, property, or the environment is not authorized. Use of this software is also subject to the terms and conditions of the Software License Agreement with CDS that you accepted at time of installation.

This manual, information, technology and software are protected by copyright law and international treaties. Unauthorized reproduction or distribution may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent of the law.

7. Index of Acronyms and Abbreviations

API	Applications Programming Interface
CDS	Constellation Data Systems
DCB	WIN32 Device Control Block
GPS	Global Positioning System
HyperTerminal	Standard Windows Communications Application
MS	Microsoft
MSDN	MS Developers Network
NMEA	National Maritime Electronics Association
RS-232	Recommended Standard 232 (from the Electronics Industry Association) for data communications
RX	Receive
SDK	Software Development Kit
TBD	To Be Described / To Be Determined
TLA	Three Letter Acronym
TX	Transmit
VSP	Virtual Serial Port
VSPAPI	Virtual Serial Port Applications Programming Interface
WIN32	Windows 32 Bit Programming Paradigm