



Network Serial Port - *Professional* Core Users Guide and Reference

For Windows XP, 2000, NT
Constellation Data Systems, Inc.
www.VirtualPeripherals.com

Copyright © 2003, 2004 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 and Overview	3
1.1	Quick Start	4
1.2	Capabilities of Network Serial Port - Professional	4
2.	Hypothetical NSP <i>Pro</i> Implementation.....	6
3.	Description of NSP <i>Pro</i> Core Components.....	7
3.1	Physical Serial Ports vs. Virtual Serial Ports	7
3.2	NSP <i>Pro</i> Utilities.....	7
3.3	NSP Device Enumerator	9
4.	Sockets and Internet Protocol Addresses.....	9
5.	NSP <i>Pro</i> Utilities	9
5.1	Syntax Conventions	10
5.2	Physical on Server to Single Utility.....	11
5.2.1	Data Flow	11
5.2.2	Command Line Parameters.....	12
5.2.3	Demonstration	13
5.3	Virtual on Server to Single Utility.....	13
5.3.1	Data Flow	14
5.3.2	Command Line Parameters.....	14
5.3.3	Demonstration	15
5.4	Physical on Client to Single Utility	15
5.4.1	Data Flow	16
5.4.2	Command Line Parameters.....	16
5.4.3	Demonstration	17
5.5	Virtual on Client to Single Utility	17
5.5.1	Data Flow	17
5.5.2	Command Line Parameters.....	18
5.5.3	Demonstration	19
5.6	<i>Add Port</i> Utility	19
5.6.1	Command Line Parameters of <i>Add Port</i>	20
5.6.2	Demonstration Using <i>Add Port</i>	20
5.7	<i>Delete Port</i> Utility	20
5.7.1	Command Line Parameters of <i>Delete Port</i>	21
5.7.2	Demonstration Using <i>Delete Port</i>	22
5.8	<i>Enum Ports</i> Utility.....	22
5.8.1	Command Line Parameters of <i>Enum Ports</i>	22
5.8.2	Demonstration Using <i>Enum Ports</i>	23
5.9	Gen System Id Utility.....	23
5.9.1	Usage of Gen System Id.....	24
6.	Installation Procedure	25
6.1	Step by Step Instructions	25
6.2	Quick Installation Verification using NSP Enumerator.....	32
7.	Detailed Installation Verification Procedure	33

8. Removal (Uninstall) Procedure	34
9. Demonstration Period	35
10. Notices	35
11. Index of Acronyms and Abbreviations	36

1. Introduction and Overview

This manual describes the Network Serial Port *Professional* (NSP *Pro*) Core components, installation, use and operation.

NSP *Pro* has the exclusive capability of expressing Virtual as well as Physical Serial Ports across a TCP/IP based Network. Using our NSP *Pro*'s Virtual Serial Port capability, ***data can be delivered directly into application software!*** This unique feature allows freedom from the clunky external cabling required by competing products.

Another benefit of the NSP *Pro*'s Virtual Serial Port capability is that there is effectively no limit on the number of Virtual Ports, and connected applications. Contrast this to competing technologies where each Network port consumes a Physical Serial Port.

The NSP *Pro* Core, as downloaded from our WebSite, is provided as a Demo. As a demo, the unit is fully functional with the exception that the Startup Screen will appear requiring you to obtain a serial number in order to prevent this screen coming up periodically. In the Demo Mode when this screen appears, you must acknowledge it by pressing on "OK". This will take you back to the License Agreement Dialog box requiring you to accept the terms and condition of the agreement. Once enabled, the Startup screen will no longer appear. See Section 6.2

Feature	NSP Pro	Competition
Moves serial port data across TCP/IP Networks	ü	ü
Access Physical Serial Ports across a Network	ü	ü
Virtual Serial Port Capability allows data delivery directly into application software.	ü	
No limit on number of Serial Ports	ü	
Software Development Kit allows custom implementations.	ü	
OEM Friendly Licensing – One time charge allows unlimited number of copies.	ü	
Robust set of COM port settings	ü	
Freedom from clunky external cables	ü	
Powerful set of Network Serial Port utilities included.	ü	
Available source code licensing	ü	

1.1 Quick Start

While it is suggested that you become thoroughly familiar with this manual, technically savvy users may wish to skim this manual in overview, and then perform the following:

1. Install the NSP *Pro* Core as per Section 6.
2. Perform the Installation Validation Procedure, described in section 7.

1.2 Capabilities of Network Serial Port - Professional

The NSP *Pro* is a product of Constellation Data Systems, Inc (CDS). The NSP *Pro* Core Product is capable of the following powerful activities:

- Access serial ports from remote systems, across a network.
- Hardware-less serial port interface emulation, with data source and endpoints to a network resource.
- Easy capture of data from a serial port data.
- Easy generation of data into a serial port.
- High speed data transfers / transmissions.
- Serial port data redirection to and from a network
- Serial port device simulation out to a network endpoint
- Dynamic Port Creation

NSP *Pro* is also easily extensible. Should you have custom requirements, the NSP *Pro Software Development Kit* (NSPSDK) is available. The NSPSDK

enables rapid development of custom NSP applications. Using the NSPSDK, NSP *Pro* becomes a powerful development accelerator, which can cut months or years from a development project!

Some examples of the custom NSP applications, developed with the NSPSDK are:

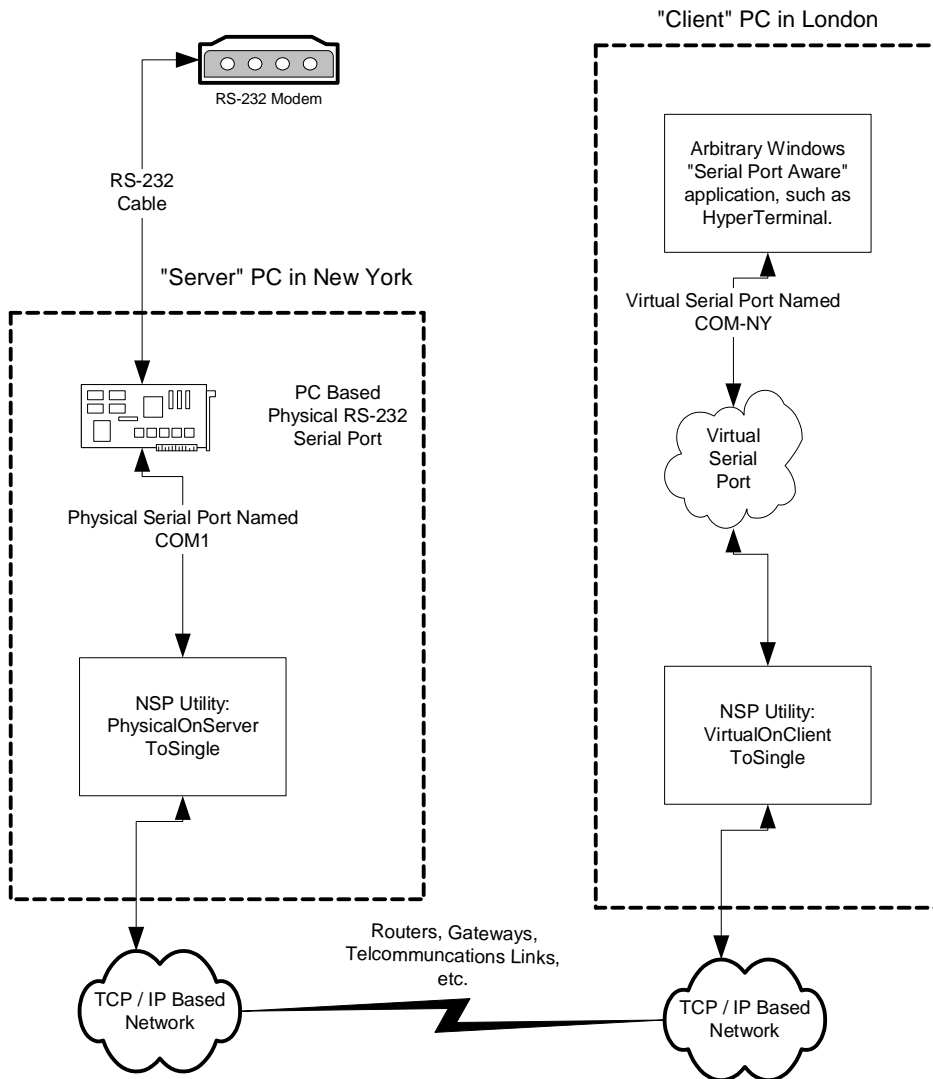
- Protocol manipulation of serial port data.
- Timing construction and reconstruction of Network generated serial port data.

Redirection of serial port data streams to/from custom-built applications and to/from network end points.

2. Hypothetical NSP *Pro* Implementation

In the following hypothetical NSP example, a serial device (in this case a modem on an RS-232 Cable) has been expressed across a network.

The user of the PC in London can access the Modem in New York, named (for example), using the serial port name: COM-NY.



3. Description of NSP *Pro* Core Components

3.1 Physical Serial Ports vs. Virtual Serial Ports

A Physical Serial Port is simply an actual serial port installed on a PC. Typically these Serial Ports conform to the RS-232 standard for serial communications, and are often simply called “RS-232 ports”. There are literally hundreds, and perhaps thousands of different types of devices (such as modems, printers, GPS units, and many more) that may be connected to a Physical Serial Port.

Note:	Windows Application software capable of accessing serial ports is referred to in this manual as “ <i>serial port aware</i> ” software. An Example of serial port aware software is HyperTerminal.
-------	---

A Virtual Serial Port, on the other hand, is a virtualized communications resource, which generally has no actual (physical) serial port hardware on the machine that hosts the serial port aware software.

3.2 NSP *Pro* Utilities

The NSP *Pro* Utilities are a powerful set of pre-built NSP applications. Many users will be able to use these utilities in stand-alone form. Other users, with more complex requirements may wish to consider using the NSP *Pro* Software Development Kit (NSPSDK) to implement custom requirements.

Note:	The NSP <i>Pro</i> Software Development Kit includes the NSP Applications Programming Interfaces (NSP API) library, and the reference designs (source code) for the NSP utilities.
-------	--

The NSP *Pro* Utilities implement network data flow using industry standard “*Socket*” based connection techniques. The system that answers a connection (like answering an incoming telephone call), is called the “*Server*” (similar to answering an incoming telephone call). Analogously, the system that initiates a connection is called the “*Client*” (similar to dialing a telephone).

<u>Utility</u>	<u>Description</u>
PhysicalOnServerToSingle.exe	This utility waits on a network Socket for a single connection. Once connected, the Physical Serial Port is bi-directionally bound to the network data traffic. Consult section 5.2 for more information.
VirtualOnServerToSingle.exe	This utility waits on a network Socket for a single connection. Once connected, the Virtual Serial Port is bi-directionally bound to the network data traffic. Consult section 5.3 for more information.
PhysicalOnClientToSingle.exe	This utility binds a Physical Serial Port through a single network Socket to a remote Server. Consult section 5.4 for more information.
VirtualOnClientToSingle.exe	This utility binds a Virtual Serial Port through a single network Socket to a remote Server. Consult section 5.5 for more information.
AddPort.exe	This utility creates a Virtual Serial Port. Consult section 5.6 for more information.
DeletePort.exe	This utility deletes a Virtual Serial Port. Consult section 5.7 for more information.
EnumPorts.exe	This utility enumerates all Virtual and Physical Serial Ports on a system. Consult section 5.8 for more information.
GenSystemId.exe	This utility displays the SYSTEM ID of computer system. Each SYSTEM ID is unique and allows CDS to identify the target machine. Consult section 5.9 for more information.

3.3 NSP Device Enumerator

Another core component of the NSP Product is the *Device Enumerator*. This enumerator is typically run at system logon time. This enumerator presents the operator with a synopsis of serial port devices, both Virtual and Physical. The Virtual devices are enumerated in a dialog box, and shown on the left hand side of that box along with the underlying VSP driver components versioning information. The Physical devices are shown on the right hand side of the dialog box, along with some additional device specific information.

After user logon, the *Device Enumerator* remains running in the background, continuously monitoring the status of the system's Virtual Serial Ports.

See section 5.9, for related information.

4. Sockets and Internet Protocol Addresses

IP addresses are typically represented in "dotted quad nomenclature", where each IP address is shown as four decimal numbers (between 0 and 255), separated by periods (e.g., "61.12.123.11"). Socket numbers are integers between 0 and 65,535. The IP address paired with the socket number comprise an NSP addressable entity.

Important	Socket numbers may be used by other applications and system components. To avoid conflicts, you may wish to consult with the Internet Naming Authority's ("INA") master lists.
-----------	--

5. NSP *Pro* Utilities

The NSP Utilities are powerful pre-built NSP applications. Many users will be able to use the NSP Utilities in stand-alone form. Other users, with more complex requirements may consider using the NSPSDK to implement custom requirements. This is available for free and can be downloaded from our website at www.VirtualPeripherals.com.

The NSP *Pro* Utilities allow Client PC's to connect and use Serial Port resources from Server PC's. The Client PC initiates the connection by attempting connection to a known socket at the Server's Internet Protocol ("IP") address.

Important Point	The "Client" PC initiates a connection much as a person dialing a telephone initiates a telephone connection. Analogously, a
-----------------	--

	<p>“Server” PC responds to connection request from a “Client” PC in a similar fashion as a person answering a telephone.</p> <p>In order for the “Server” to answer an incoming “Client” connection, an NSP “Server” type utility must be running at the time the connection is attempted.</p>
--	--

5.1 Syntax Conventions

Syntax represents the command line format you must follow when using the NSP *Pro* utilities (see section 5). Unless otherwise indicated, you may enter commands and parameters in either upper or lower case. It is expected that you type the ENTER key at the end of a utility command.

Consider the following mapping:

Utility parameter1 {parameter 2|parameter2a} [parameter 3a | parameter 3b] ... parameter 'n' [switches]

<u>Element</u>	<u>Usage</u>
<i>Utility</i>	The name of the utility command (VirtualToVirtual, for example).
[]	Brackets indicate that an item is optional.
{ }	Braces indicate a choice of parameters, where one of the choices is mandatory.
	Separates two or more mutually exclusive choices.
...	Indicates that the preceding parameter may be repeated zero or more times.
[switches]	Zero or more optional command switches. A switch begins with a slash, “/p”, for example.

In the preceding Syntax mapping;

- ‘Utility’ is the name of the utility Command (this is what you enter on the command line to run the corresponding utility).
- Parameter 1 is mandatory
- Parameters 2a or 2b are mandatory, and are mutually exclusive.
- Parameters 3a and 3b are optional, and when specified are mutually exclusive.
- Additional parameters may follow the optional parameter 3, and are may be considered as parameters ... n-2, n-1, n.
- Optional command line switches as required.

5.2.2 Command Line Parameters

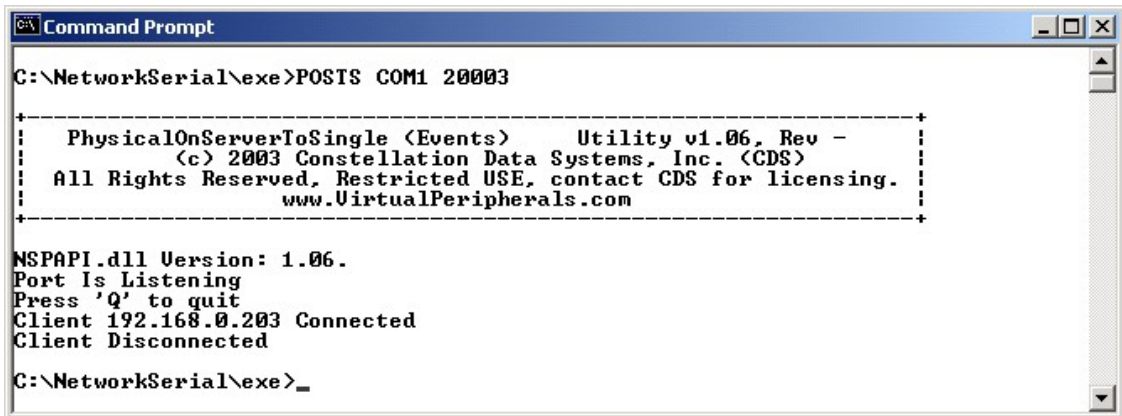
The utility may be run as either “*PhysicalOnServerToSingle.exe*”, which is rather verbose, or more simply as “*posts.exe*”.

Consider the following syntax mapping:

Usage:	<pre>Posts.exe Physical SocketNumber [switches ...] Physical ----- Physical serial port (COM1, COM2, etc.) SocketNumber -- Number of the Socket on the host machine [switches] ---- Optional settings of the physical port, etc. (see below).</pre>
Runtime	<pre>'Q' then 'ENTER' typed on the keyboard causes the utility to quit (exit).</pre>
[switches]	<pre>Optional settings of the physical port, etc. Note: All VALUE's are in decimal. /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} /InX:{ENABLE DISABLE} /XonLim:VALUE /XoffLim:VALUE /XonChar:VALUE /XoffChar:VALUE /NullDiscard{ENABLE DISABLE} /RtsControl:{DISABLE ENABLE HANDSHAKE TOGGLE} /ErrorReplace:{ENABLE DISABLE} /ErrorReplaceChar:VALUE /ByteSize:VALUE /ReadIntervalTimeout:VALUE /MonitorHex /MonitorAscii</pre>

5.2.3 Demonstration

The following hypothetical example shows a usage where the data traffic to and from physical port, COM1, is mapped to socket 20003 on the host machine (which will act in “server” mode).



```
Command Prompt
C:\NetworkSerial\exe>POSTS COM1 20003

+-----+
| PhysicalOnServerToSingle (Events)      Utility v1.06, Rev - |
| (c) 2003 Constellation Data Systems, Inc. (CDS) |
| All Rights Reserved, Restricted USE, contact CDS for licensing. |
| www.VirtualPeripherals.com |
+-----+

NSPAPI.dll Version: 1.06.
Port Is Listening
Press 'Q' to quit
Client 192.168.0.203 Connected
Client Disconnected

C:\NetworkSerial\exe>_
```

The utility will continue to answer connections, showing those connections as they are made and as they end (terminated or broken). Execution continues until the ‘Q’ keystroke is pressed. Observe above that the utility displays on the console, the IP address of the connected client, as those connections are made.

At the time of connection this utility sends the following message to the client system, through the data stream:

<Connected Remote – PhysicalOnServerToSingle>

When a disconnection is observed, the <Client has Disconnected> message will be observed in the receive data stream. This message is inserted locally by the utility into the receive data stream, and is NOT generated on the remote system.

5.3 Virtual on Server to Single Utility

This utility allows a bi-directional data stream, to and from a Network Socket, to be attached to a Virtual Serial Port. Note that Virtual Serial Ports are named at the time of software installation. Serial Port “aware” applications (HyperTerminal, for example), may connect to the corresponding Virtual Serial Port.

This utility will “answer” connections (one connection active at a time – “single”). The utility continues to run, observing socket connections and disconnections, until a ‘Q’ keystroke is observed on the console.

5.3.1 Data Flow

(Serial Port “aware” application) ó (Virtual Serial Port) ó
VirtualOnServerToSingle Utility ó (IP Address / Socket)

5.3.2 Command Line Parameters

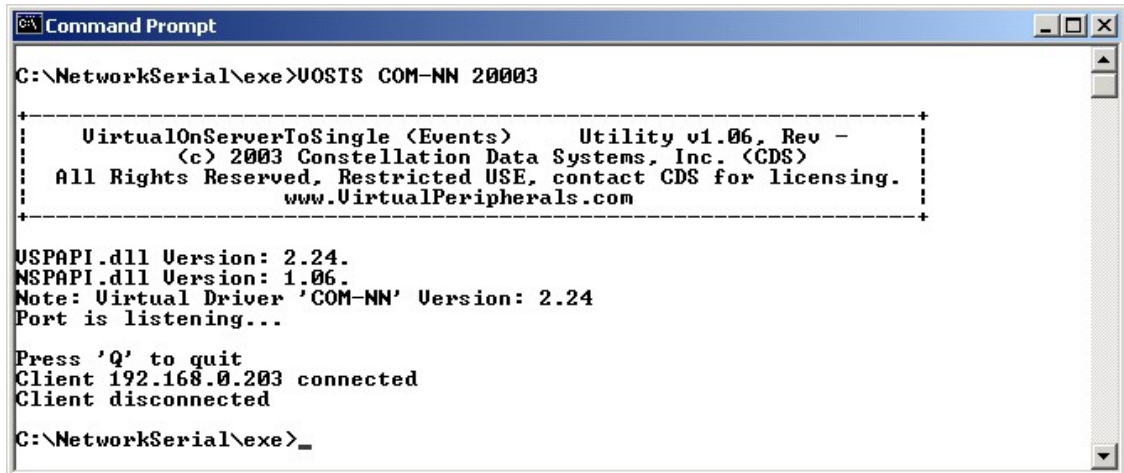
The utility may be run as either “*VirtualOnServerToSingle.exe*”, which is rather verbose, or more simply as “*vosts.exe*”.

Consider the following syntax mapping:

Usage:	<pre>VirtualOnServerToSingle VspName SocketNumber [switches] VspName ----- Name of Virtual Serial Port SocketNumber --- Socket number (on local machine).</pre>
[switches]	<pre>/INFO ----- Show DLL information. /MonitorHex ---- Show data traffic in HEX on the console. /MonitorAscii -- Show data traffic in ASCII on the console. /ReadIntervalTimeout:VALUE -- Timeout value between byte reads. (Default:100 - ms)</pre>

5.3.3 Demonstration

The following hypothetical example shows a usage where the data traffic to and from Virtual Serial Port (named "COM-NN"), is mapped to socket 20003 on the host machine (which will act in "server" mode).



```
Command Prompt
C:\NetworkSerial\exe>UOSTS COM-NN 20003

-----
:   VirtualOnServerToSingle (Events)   Utility v1.06, Rev -
:   (c) 2003 Constellation Data Systems, Inc. (CDS)
:   All Rights Reserved, Restricted USE, contact CDS for licensing.
:   www.VirtualPeripherals.com
-----

USPAPI.dll Version: 2.24.
NSPAPI.dll Version: 1.06.
Note: Virtual Driver 'COM-NN' Version: 2.24
Port is listening...

Press 'Q' to quit
Client 192.168.0.203 connected
Client disconnected

C:\NetworkSerial\exe>_
```

The utility will continue to answer connections, showing those connections as they are made and as they end (terminated or broken). Execution continues until the 'Q' keystroke is pressed. Observe above that the utility displays on the console, the IP address of the connected client, as those connections are made.

At the time of connection, this utility sends the following message to the client system, through the transmit data stream:

<Remote Connected – *VirtualOnServerToSingle*>

When a disconnection is observed, the following message is inserted locally by this utility into the receive data stream:

<Client Disconnected>

5.4 Physical on Client to Single Utility

This utility allows a bi-directional data stream, to and from a Network Socket, to be attached to a Physical Serial Port on the local ("client") machine. Physical devices such as serial modems, serial printers, and so forth may then be attached to the Physical Serial Port.

This utility will connects to a remote machine (one connection active at a time – "single"). The utility continues to run, until a socket disconnection is observed,

or until a 'Q' keystroke is observed on the console, at which time the utility is terminated.

5.4.1 Data Flow

(Serial Port device, such as a modem) ó (Physical Serial Port) ó
PhysicalOnClientToSingle Utility ó (IP Address / Socket).

5.4.2 Command Line Parameters

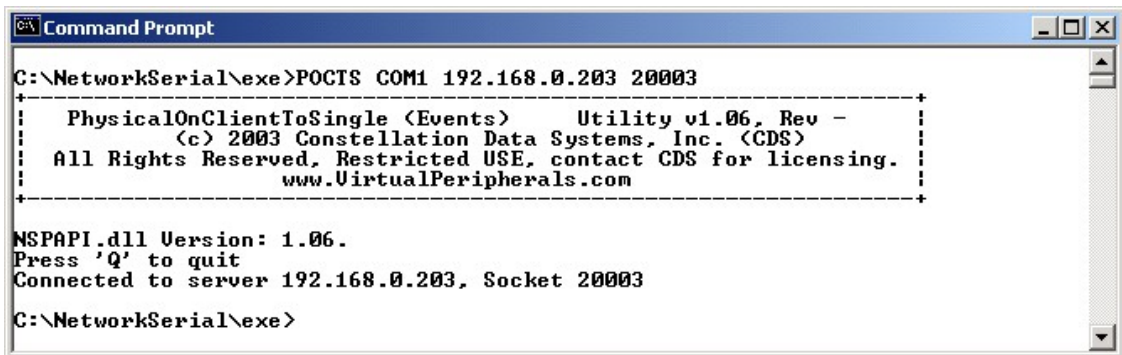
The utility may be run as either "*PhysicalOnClientToSingle.exe*", which is rather verbose, or more simply as "*pocts.exe*". Consider the following syntax mapping:

Usage:	<pre>pocts Physical NetAddress SocketNumber [switches...] Physical ----- Physical serial port, such as COM1, or COM2, etc. NetworkAddress - Network address of the server machine, such as 199.1.2.44 SocketNumber --- Number of the Socket on the host machine, an integer, such as 10023 [switches] ----- Optional settings of the physical port, etc.</pre>
[switches]	<pre>Note: All VALUE's are in decimal. /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} /InX:{ENABLE DISABLE} /XonLim:VALUE /XoffLim:VALUE /XonChar:VALUE /XoffChar:VALUE /NullDiscard{ENABLE DISABLE} /RtsControl:{DISABLE ENABLE HANDSHAKE TOGGLE} /ErrorReplace:{ENABLE DISABLE} /ErrorReplaceChar:VALUE /ByteSize:VALUE</pre>

	<pre>/ReadIntervalTimeout:VALUE /MonitorHex /MonitorAscii</pre>
--	---

5.4.3 Demonstration

The following hypothetical example shows a usage where the data traffic to and from Physical Serial Port (“COM1”), is connected to socket 20003 on the remote machine, which will act in “server” mode.



```
Command Prompt  
C:\NetworkSerial\exe>POCTS COM1 192.168.0.203 20003  
-----  
PhysicalOnClientToSingle <Events>      Utility v1.06, Rev -  
      <c> 2003 Constellation Data Systems, Inc. <CDS>  
All Rights Reserved, Restricted USE, contact CDS for licensing.  
      www.VirtualPeripherals.com  
-----  
NSPAPI.dll Version: 1.06.  
Press 'Q' to quit  
Connected to server 192.168.0.203, Socket 20003  
C:\NetworkSerial\exe>
```

The utility will terminate at the time a disconnection is observed. The utility may also be terminated when a ‘Q’ keystroke is pressed on the console.

At the time of connection, the following message is sent by this utility in the transmit data stream:

<Remote Connected – PhysicalOnClientToSingle>

5.5 Virtual on Client to Single Utility

This utility allows a bi-directional data stream, to and from a Network Socket, to be attached to a Virtual Serial Port. Note that Virtual Serial Ports are named at the time of software installation. Serial Port “aware” applications (HyperTerminal, for example), may connect to the corresponding Virtual Serial Port.

This utility will connects to a remote machine (one connection active at a time – “single”). The utility continues to run, until a socket disconnection is observed, or until a ‘Q’ keystroke is observed on the console, at which time the utility is terminated.

5.5.1 Data Flow

(Serial Port “aware” application) ○ (Virtual Serial Port) ○

VirtualOnClientToSingle Utility ó (IP Address / Socket)

5.5.2 Command Line Parameters

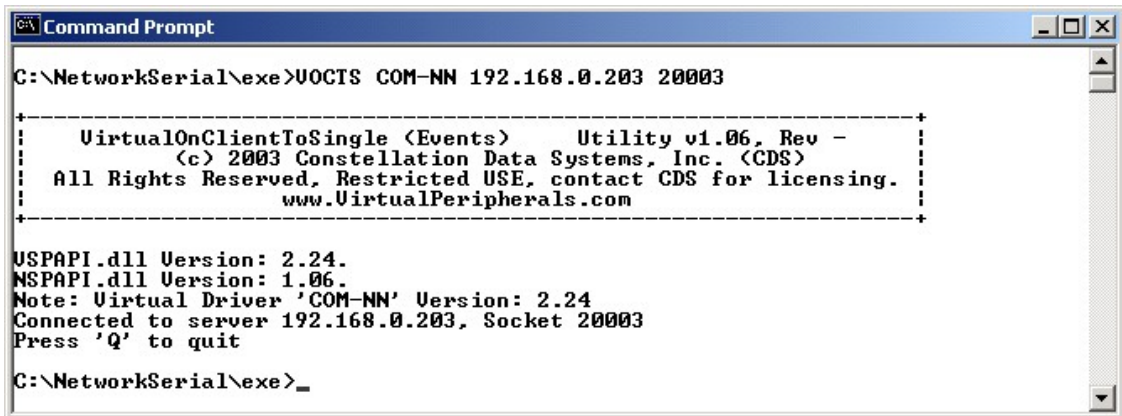
The utility may be run as either “*VirtualOnClientToSingle.exe*”, which is rather verbose, or more simply as “*vocts.exe*”.

Consider the following syntax mapping:

Usage:	<pre>VirtualOnClientToSingle VspName NetworkAddr SocketNumber [switches...] VspName ----- Name of Virtual Serial Port NetworkAddr ---- Network (IP) address of remote machine. SocketNumber --- Socket number which remote machine will answer (connect).</pre>
[switches]	<pre>/INFO ----- Show DLL information. /MonitorHex ---- Show data traffic in HEX on the console. /MonitorAscii -- Show data traffic in ASCII on the console. /ReadIntervalTimeout:VALUE -- Timeout value between byte reads. (Default:100 - ms)</pre>

5.5.3 Demonstration

The following hypothetical example shows a usage where the data traffic to and from Virtual Serial Port (named "COM-NN"), is mapped to socket 20003 on IP address 192.168.0.203.



```
Command Prompt
C:\NetworkSerial\exe>UOCTS COM-NN 192.168.0.203 20003
-----+-----
|   VirtualOnClientToSingle (Events)   Utility v1.06, Rev -
|   (c) 2003 Constellation Data Systems, Inc. (CDS)
|   All Rights Reserved, Restricted USE, contact CDS for licensing.
|   www.VirtualPeripherals.com
|-----+-----

USPAPI.dll Version: 2.24.
NSPAPI.dll Version: 1.06.
Note: Virtual Driver 'COM-NN' Version: 2.24
Connected to server 192.168.0.203, Socket 20003
Press 'Q' to quit
C:\NetworkSerial\exe>_
```

Execution continues until the 'Q' keystroke is pressed, or a disconnection occurs. Observe above that the utility displays on the console, the IP address and socket number of the connected server.

At the time of connection, this utility sends the following message to the client system, through the transmit data stream:

<Remote Connected – *VirtualOnClientToSingle*>

When a disconnection is observed, the following message is inserted locally by this utility into the receive data stream:

<Client Disconnected>

5.6 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. Avoid already defined port names, or common port names such as COM1 or COM2.

If a port with the selected name already exists, a message will appear which explains why the Virtual Serial Port could not be created.

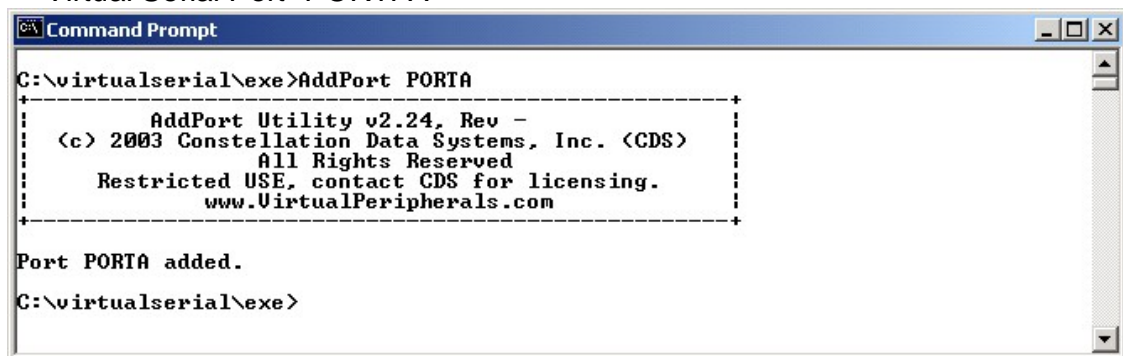
5.6.1 Command Line Parameters of *Add Port*

Consider the following syntax mapping:

Syntax	AddPort PortA
Parameters	PortA – Name of the Virtual Serial Port that is to be created.
Switches	<i>/?</i> – Prints command line usage.

5.6.2 Demonstration Using *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 PORTA
-----
AddPort Utility v2.24, Rev -
(c) 2003 Constellation Data Systems, Inc. (CDS)
All Rights Reserved
Restricted USE, contact CDS for licensing.
www.VirtualPeripherals.com
-----
Port PORTA added.
C:\virtualserial\exe>
```

5.7 Delete Port Utility

The *Delete Port* (“deleport.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 PORTA
+-----+
| DeletePort Utility v2.24, Rev - |
| (c) 2003 Constellation Data Systems, Inc. <CDS> |
| All Rights Reserved |
| Restricted USE, contact CDS for licensing. |
| www.VirtualPeripherals.com |
+-----+
USPAPI.dll Version: 2.24.
Driver Version: 2.24.
Error : Can not delete port PORTA...port is in use.
C:\virtualserial\exe>
```

To clear this condition, free the device, in this case by simply removing the connection to the Virtual Serial Port.

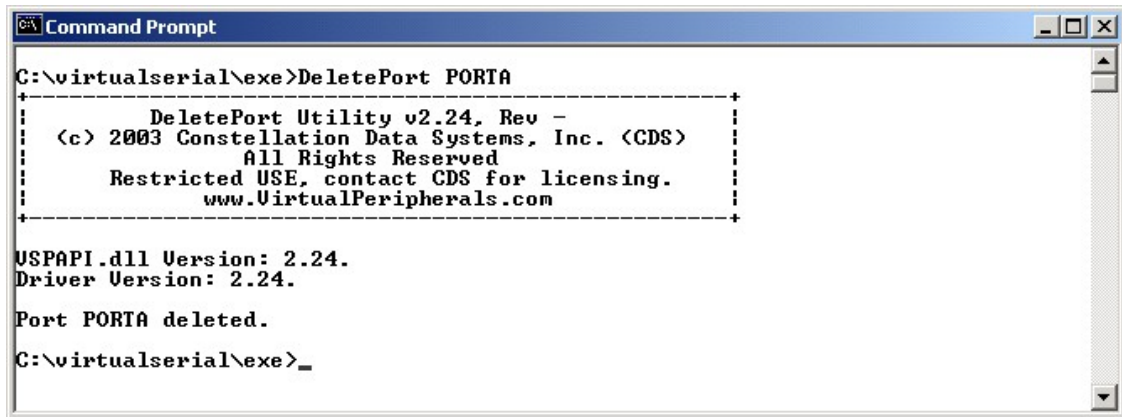
5.7.1 Command Line Parameters of *Delete Port*

Consider the following syntax mapping:

Syntax	DeletePort PortA
Parameters	PortA – Name of the Virtual Serial Port that is to be deleted.
Switches	<i>/?</i> – Prints command line usage.

5.7.2 Demonstration Using *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 PORTA
+-----+
| DeletePort Utility v2.24, Rev - |
| (c) 2003 Constellation Data Systems, Inc. (CDS) |
| All Rights Reserved |
| Restricted USE, contact CDS for licensing. |
| www.VirtualPeripherals.com |
+-----+
USPAPI.dll Version: 2.24.
Driver Version: 2.24.
Port PORTA deleted.
C:\virtualserial\exe>
```

5.8 *Enum Ports* Utility

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

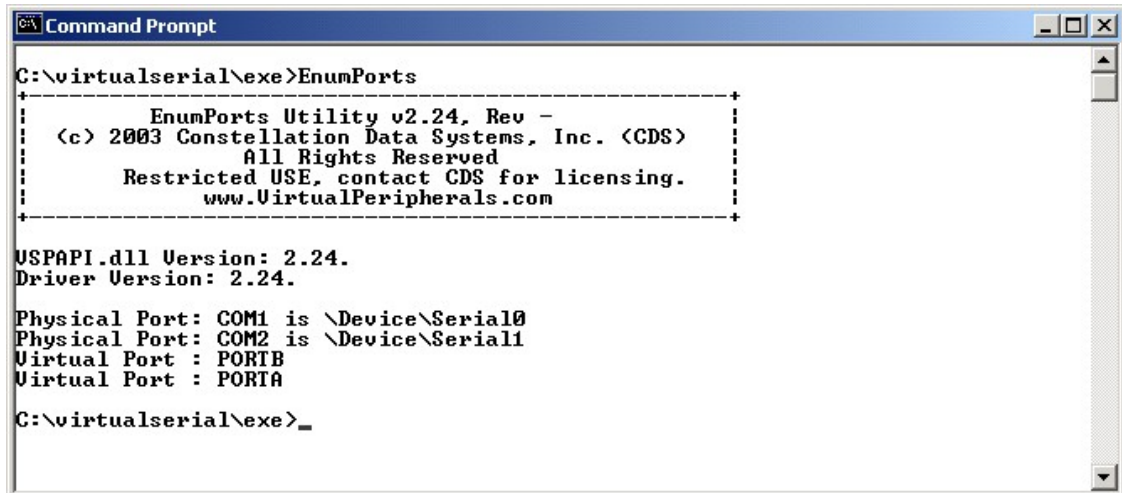
5.8.1 Command Line Parameters of *Enum Ports*

Consider the following syntax mapping:

Syntax	EnumPorts
Parameters	This utility does not take any parameters.
Switches	This utility does not have any options available.

5.8.2 Demonstration Using *Enum Ports*

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



```
Command Prompt
C:\virtualserial\exe>EnumPorts
+-----+
| EnumPorts Utility v2.24, Rev - |
| (c) 2003 Constellation Data Systems, Inc. (CDS) |
| All Rights Reserved |
| Restricted USE, contact CDS for licensing. |
| www.VirtualPeripherals.com |
+-----+
USPAPI.dll Version: 2.24.
Driver Version: 2.24.
Physical Port: COM1 is \Device\Serial0
Physical Port: COM2 is \Device\Serial1
Virtual Port : PORTB
Virtual Port : PORTA
C:\virtualserial\exe>
```

5.9 Gen System Id Utility

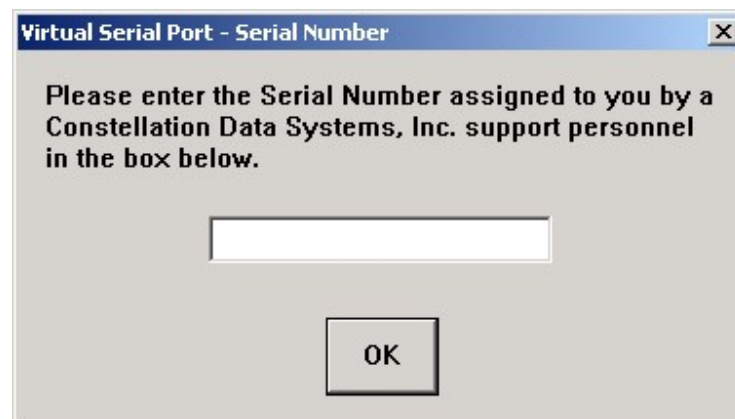
The *Gen System Id* utility generates a System ID for the computer it is ran on. Every computer system has a unique System ID. The System ID is needed by CDS to assign a Serial Number for the NSP, which will enable properly licensed NSP software to run on the target system. The System ID and *Gen System Id* utility are discussed in the context of the installation procedure, on page 26 of this manual.



5.9.1 Usage of Gen System Id

Besides being able to generate a System ID if needed, *Gen System Id* also displays the System ID and Serial Number of the computer system. If the Virtual Serial Port application is running in DEMO mode, Gen System Id allows the user to enter a Serial Number, which is assigned to the desired computer system by a CDS representative.

To enter a Serial Number, simply click "File" in the menu bar, and select "Enter Serial Number". A separate "Serial Number" dialog box appears with some simple instructions and an area to enter a Serial Number.

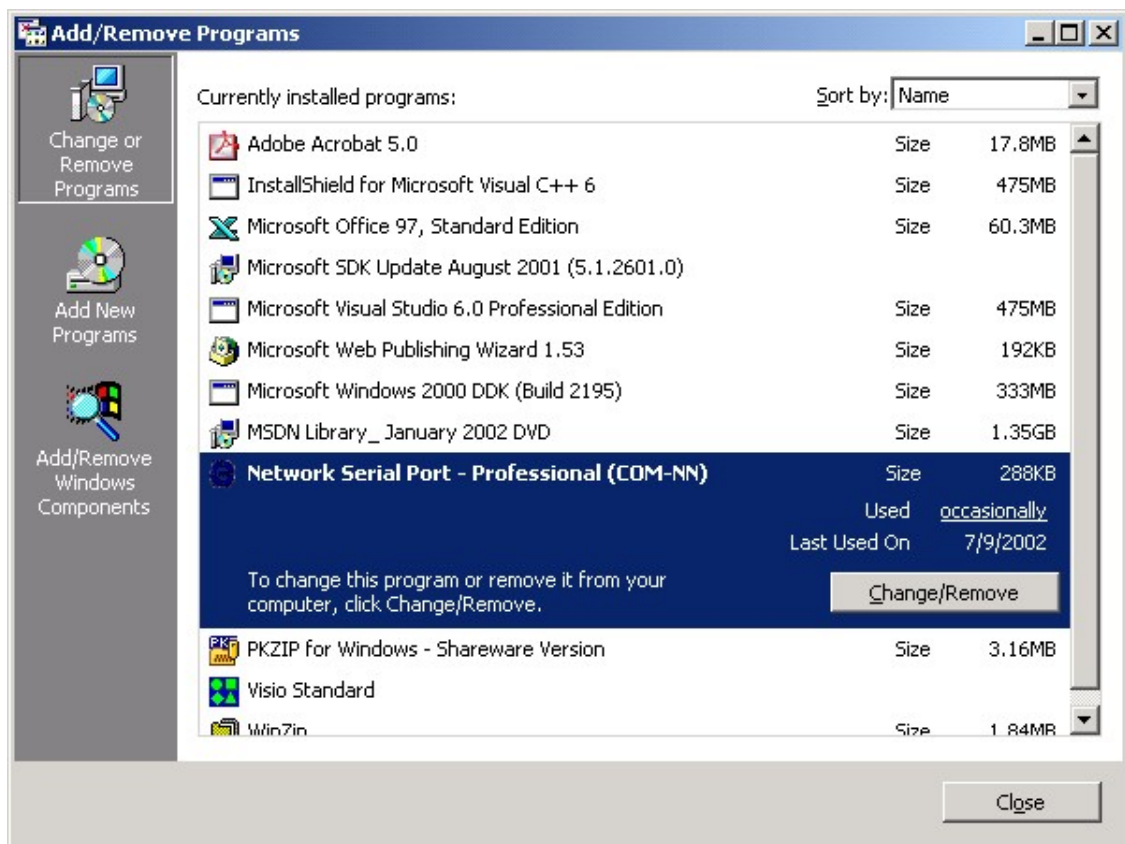


6. Installation Procedure

6.1 Step by Step Instructions

It is recommended that the NSP *Pro* be installed on a technically correct installation of Windows. Each installation of NSP *Pro* will enable one Virtual Serial Port. To create more than one Virtual Serial Ports, simply install NSP *Pro* one or more times. NOTE: Virtual Serial Ports can be created after the installation process via the AddPort Utility (see section 5.6 for more details).

1. **Log onto the target machine as the system administrator. You must be logged on as the system administrator in order to correctly install NSP Pro.**
2. Remove any existing NSP *Pro* installations of a version prior to the target version. While multiple NSP *Pro* installations may be made on a single PC, it is recommended that they all be of the same version. Go to the Control Panel function; Add/Remove programs, and identify any NSP installations of prior version. Consider the following "Add/Remove Programs" dialog from a Windows 2000 installation:



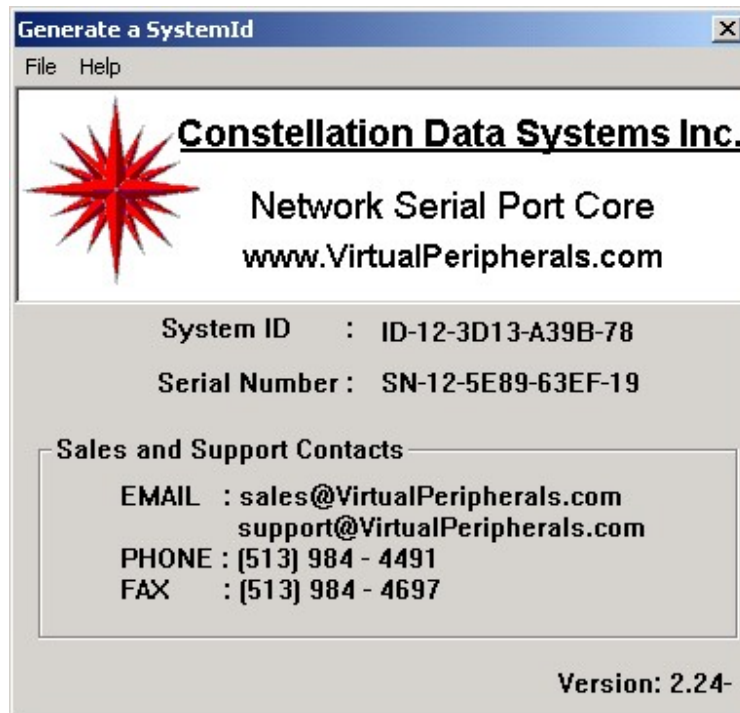
In this example there is a single NSP *Pro* installation with a Virtual Port named “COM-NN”. Each Virtual Port installation will have a separate line item on this dialog. Of course the port names will differ on your system. Remove all prior installations NSP *Pro* those from the target system, then reboot the target machine, and begin again with step #1.

3. The setup executable is named using the following nomenclature: “NspProCoreXXX.exe”, where XXX is a short version description embedded in the file name. For example, version 1.06 of the NSP *Pro* Core, XXX would be 106, and the entire module would be named “NspProCore106.exe”.

Run the setup executable. You will be prompted for the “Unzip To Folder”. The folder “c:\NspPro” is recommended. The NSP *Pro* Core installation files will then be unzipped into that directory. The NSP *Pro* Core installation file set consists of the files shown in the following table. Note that while the modules may vary from one version to another version of the software, the following modules should generally be present in the NSP *Pro* installation directory (or installation disk).

<p>NspStartup.exe Os.dat _ISDEL.EXE SETUP.EXE Lang.dat _INST32I.EX_ _setup.dll setup.ins _sys1.cab DATA.TAG SETUP.INI _user1.cab layout.bin data1.cab setup.lid Vsp.sys License.txt Connector.bmp</p>	<p>NSP <i>Pro</i> Core installation files, and install shield information.</p>
<p>Exe\VirtualOnServerToSingle.exe Exe\PhysicalOnServerToSingle.exe Exe\VirtualOnClientToSingle.exe Exe\PhysicalOnClientToSingle.exe Exe\Addport.exe Exe\Deletport.exe Exe\Enumports.exe Exe\GenSystemId.exe</p>	<p>These are the NSP utilities. Consult Section 5 for more information</p>

4. Run the “exe\GenSystemId.exe” utility (see above). This will display the System ID number.

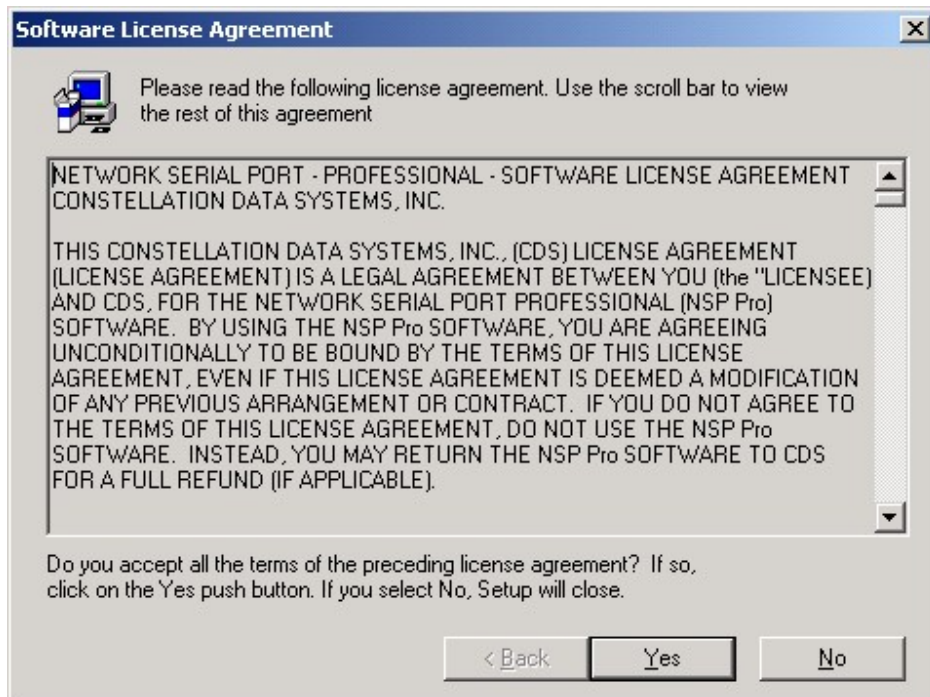


Users can then provide this number to CDS support and a Serial number will be assigned for this system. This serial number will be used later in the installation procedure, which will fully enable the NSP on this system.

5. Run “setup.exe” from the installation directory (generally “C:\NspPro”). From the START button, select RUN, and then “c:\NspPro\Setup.exe”.

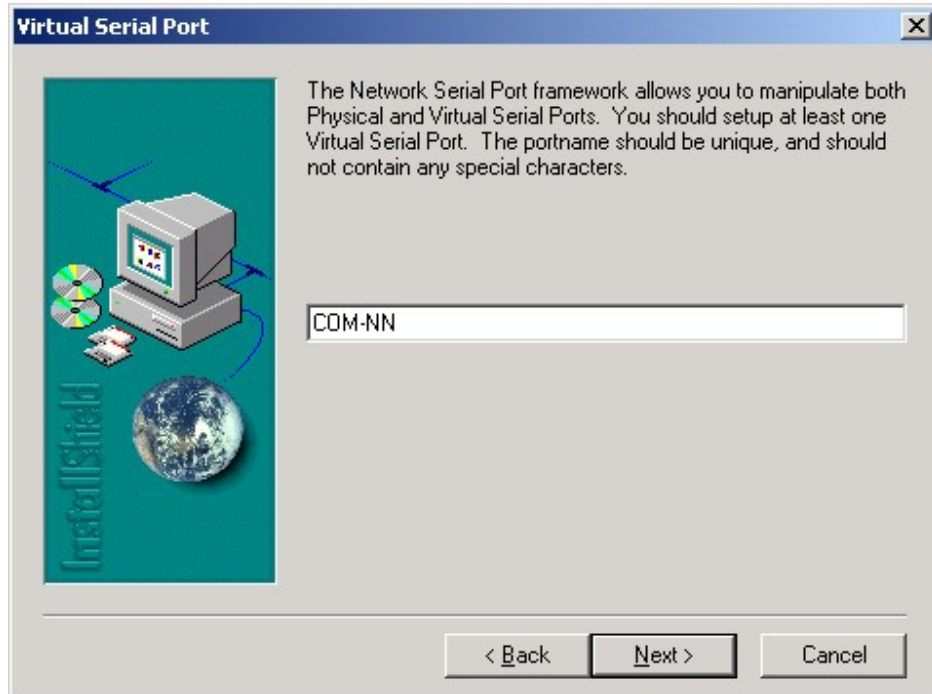
Should the installation device be a floppy disk, simply select the START button, then RUN “a:setup”.

6. The NSP Software License agreement will then be displayed for your acceptance.



After reading the entire agreement, if you accept the Software License agreement's terms, indicate acceptance by selecting "Yes", Should appear. Should you not be agreement with the terms, select "No", and the installation procedure will terminate.

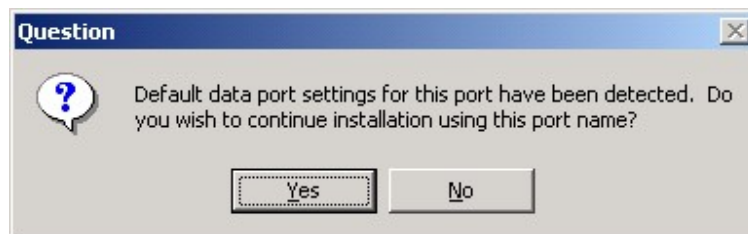
7. After the Software License dialog is displayed and accepted, the Install Shield Virtual Serial Port Dialog will be displayed:



Enter the name of the Virtual Serial Port that you would like to create.

This will be the same name that applications, such as HyperTerminal use to identify the VSP device. Avoid already defined port names, or common port names such as COM1 or COM2. Once a port name is chosen, such as ("Virtual1"), select "Next". The "Enter Product Serial Number" dialog box should then appear.

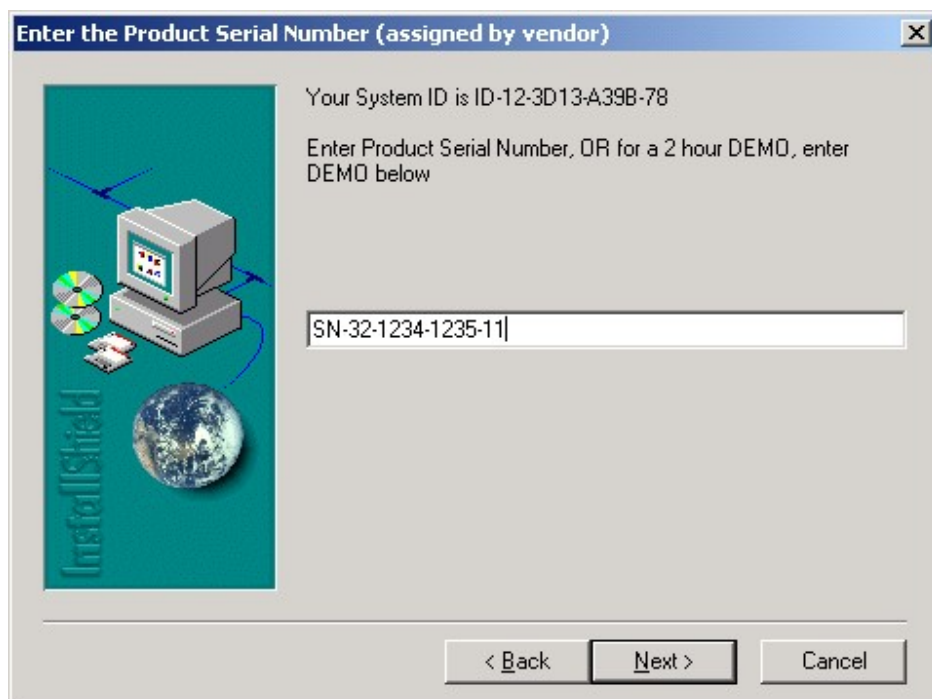
If a port with the selected name already exists, then a message box will appear which prompts you to select another name. You may also observe the following dialog:



This informative dialog simply indicates that a previous installation (using that port name) was detected, and that some residual port data still remains in the

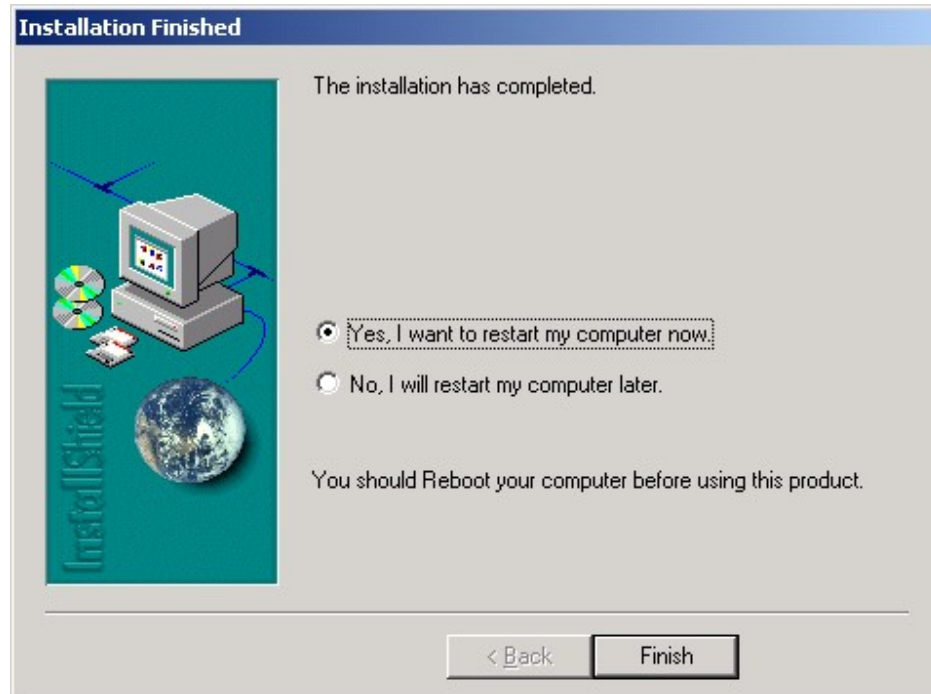
registry. If the port name is correct, simply select YES, otherwise select NO and re-enter the port name.

8. The “Enter Product Serial Number” will allow you to enter the product serial number assigned to you by CDS. Users running the product for Demonstration purposes can simply enter “DEMO”. Should you choose to enter a Serial Number at a later date, you can do so through the *Gen System Id* Utility; please refer to Section 5.9. Note: Product Serial Numbers should be protected in accordance with the terms of the Software License Agreement.



Once you enter a valid product serial number, the file copy phase of installation procedure will begin. You should then observe progress bars showing files being written to the disk. Rather quickly the “Installation Finished” dialog box should appear.

9. The “Installation Finished” dialog box (shown below), prompts you to reboot the machine in order complete the installation process.

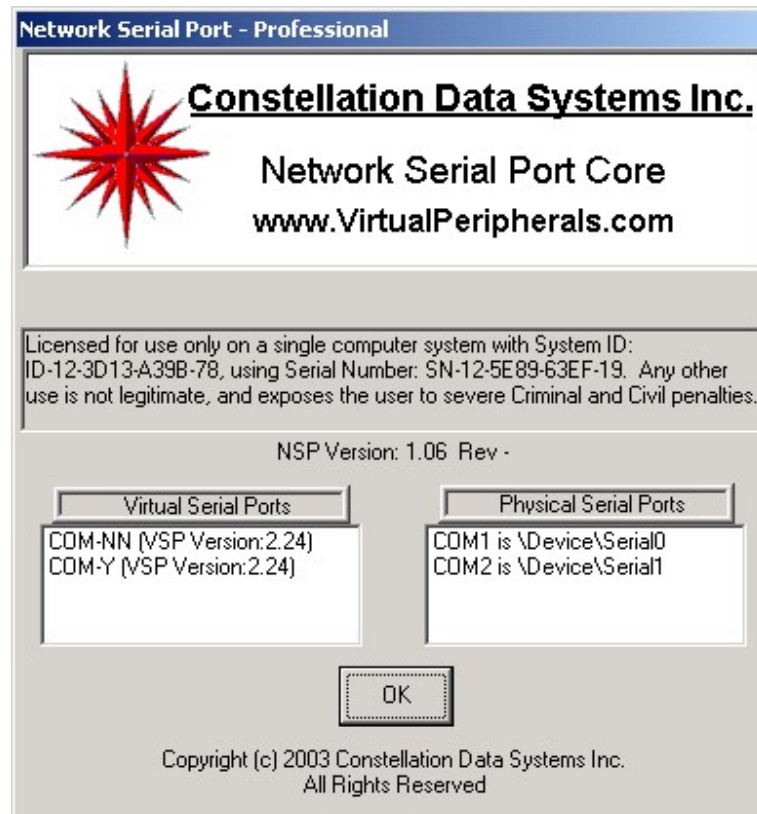


If at this time you wish to add another Virtual Serial Port, select the “No” option button and then select “Finish”. Repeat steps #7 through #9 in order for another Virtual Serial Port to be installed. Once all ports have been installed, it will be necessary to reboot the target machine.

Virtual Serial Ports can also be dynamically created outside of the Installation Procedure via the *Add Port* Utility. A feature of adding ports in this manner is that the target machine does not need to be rebooted in order for the newly installed Virtual Serial Ports to be operable. For more information, please refer to Section 5.6.

6.2 Quick Installation Verification using NSP Enumerator

Following reboot after successful installation, the following dialog box should appear at logon time. This dialog is referred to as the “Port Enumerator”.



Note that this dialog enumerates all installed and functioning Virtual Serial Ports, as well as their version number. While it is possible that different versions of the VSP may be installed on the same system, configurations with mixed versions are not suggested. Review the VSP Device Enumerator after installation and verify that all Virtual Serial Ports are of the same version. Also verify that the “Startup Version” is of the same version number.

The physical serial devices and resources, which are generally either RS-232 ports or physical modems, are shown on the right hand side of the dialog box. Should you desire to see this information at any time after boot, it may be run from the “Start” bar, to “Programs”, and then “Startup”.

Note: an error 31 shown after the port name indicates that the demonstration period has expired. Simply reboot to continue software demonstration for another period.

7. Detailed Installation Verification Procedure

The following procedure may be performed to validate the NSP *Pro* core framework. You should be logged on as the system administrator to perform this procedure.

1. Connect 2 PC's with dedicated Ethernet capability. Configure both with TCP/IP, and specify each with its own static IP addresses. For the purpose of this procedure IP addresses 199.1.1.1, and 199.1.1.2 are suggested.
2. Use the standard command line utility "ping" to verify connections between the 2 systems. For example to "ping" a remote system with IP address 199.1.1.1, simply type "ping 199.1.1.1" on the command line.
3. Install the NSP *Pro* Core as per section 6; install a Virtual Serial Port named "COM-V", on both systems.
4. Start HyperTerminal on both systems, and configure HyperTerminal to COM-V. Note: HyperTerminal settings should be saved to disk and HyperTerminal should then be reloaded. This is necessary as the new settings generally are not used except on HyperTerminal startup.
5. On the PC with IP address 199.1.1.1, start a "command prompt" window, and issue the following command line utility:
`VirtualOnServerToSingle COM-V 2003`
6. On the PC with IP address 199.1.1.2, start a "command prompt" window, and issue the following command line utility:
`VirtualOnClientToSingle COM-V 199.1.1.1 2003`
7. You should observe connection messages on both systems in the serial data stream (in the HyperTerminal window). You should also see connection messages on the consoles of both "command prompt" windows.

8. Removal (Uninstall) Procedure

Go to the Control Panel, and select the “Add/Remove Programs” function. You should observe an NSP *Pro* entry for each defined Virtual Serial Port. Select the VSP you wish to remove. A reboot after removal is recommended. Upon reboot the Port Enumerator should no longer enumerate the port just removed.

Note: When the last NSP *Pro* is removed from a system, the “Virtual Serial Port Startup Dialog” is no executed.

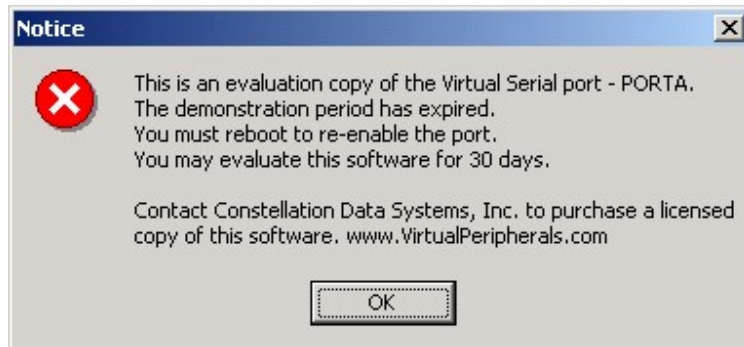
<p>Errata NT4 - Install shield has been observed to not load uninstall data on certain Windows NT4 systems.</p>	<p>To remove the NSP <i>Pro</i> from affected systems use the following procedure:</p> <ol style="list-style-type: none">1. Log on as the system administrator.2. Use REGEDIT to remove all “VirtualSerial” sub keys from HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services3. Then right click on the START button and “explore all users”. Then delete all Network Serial Port - Professional from the “Programs”, “Startup”.4. Remove the installation directory; c:\NspPro.5. Reboot the system for the changes to take affect.
---	---

The registry key “HKEY_LOCAL_MACHINE\SOFTWARE\Constellation Data Systems” will need to be removed if it exists.

Remove all Virtual Serial Port entries in:
“HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\SERIALCOMM”.

9. Demonstration Period

The Virtual Serial Port provided for demonstration (“DEMO”) disables itself several hours following boot. Logons with disabled serial ports will cause the dialog box below to appear, and then those disabled ports will be enumerated at startup with “error 31”.



Should continued evaluation be desired, simply reboot the target machine, and several hours of additional evaluation will be possible.

10. 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.

11. Index of Acronyms and Abbreviations

API	Applications Programming Interface
BPS	Bits per Second ("baud")
CDS	Constellation Data Systems
DOS	Disk Operating System
DLL	Dynamic Link Library
GPS	Global Positioning System
HyperTerminal	Standard Windows Communications Application
IP	Internet Protocol
MS	Microsoft
MSDN	MS Developers Network
NSP	Network Serial Port
OEM	Original Equipment Manufacturer
PCR	Physical Communications Resource (Such as a UART)
RX	Receive
SDK	Software Development Kit
TLA	Three Letter Acronym
TX	Transmit
UART	Universal Asynchronous Receiver / Transmitter (Hardware)
VSP	Virtual Serial Port
VSPAPI	Virtual Serial Port Applications Programming Interface
WIN16	Windows 16 Bit Programming Paradigm (Arguably Obsolete)
WIN32	Windows 32 Bit Programming Paradigm
XON	Transmit On
XOFF	Transmit Off