VPortPro V2.4 - Online Help manual

content | next >

1. About This Manual

This User's Manual describes how to install, configure and use the VPortPro package on a 32-bit IBM PC or compatible personal computer running one of the following operating systems: MS Windows 2K/2K3/XP/XP-x64. A small volume of the manual reflects simplicity of using this software tool created nevertheless on the basis of up-to-date information technologies.

The following items will be covered:

- Purpose and composition of VPortPro
- Hardware & Software requirements of VPortPro
- Installation procedure for VPortPro
- Configuring VPortPro
- VPortPro working sessions.

content | next >

< previous | content | next >

2. Introducing to VPortPro

The VPortPro package contains two main programs: **VCOM-TCP Primary Client** and **COM Port Redirector**.

Both the programs make serial (RS232) data from serial ports on your PC available on TCP/IP-based networks and make TCP/IP data available on serial ports of your PC. The software can be used to pass serial data across a corporate intranet or over the Internet.

The programs support two transport data stream protocols: Raw Connection and Telnet (with the COM Port Control features specified by IETF RFC 2217).

You could attach a serial device to a COM port on your PC, run **COM Port Redirector** and then connect to a serial device from any other PC on your network. This would allow you to use Telnet or any other TCP/IP communications software to read or write to the serial device directly from any PC located on the same network.

VCOM-TCP Primary Client functions as a TCP/IP client (for both the protocol types).

COM Port Redirector can be configured to function as a TCP/IP server and as a TCP/IP client (for both the protocol types). **VCOM-TCP Primary Client** is fully compatible with **COM Port Redirector** configured for the Client mode.

Also both the programs can create virtual serial ports on your PC and redirect them to TCP/IP ports and vice versa. These allow PC applications to communicate with any server that accepts a TCP/IP socket connection, including another PC running **VCOM-TCP Primary Client** or **COM Port Redirector**.

Applications running on your PC (with the VPortPro's programs running) can connect to a remote serial port by connecting to a remote TCP/IP port. This allows the applications (if configured) to use either local serial ports or remote serial ports (e.g., on serial servers).

The VPortPro's programs support full two-way I/O for full control of serial instruments (i.e. any instrument that you can plug into the serial port (RS232, RS422 or RS485) of your PC). Typical devices include most serial devices like gages, scales and balances, sensors, micrometers, calipers, bar code scanners, densitometers, laboratory instruments, electronic balance or electronic measuring instrument, RF equipment, communication systems, etc.

The COM Port Control protocol specified by IETF RFC 2217 can be used for any serial device attached to an access server. Such devices could be: serial printers, plotters, monitoring devices such as pipe line monitors or medical monitors, general office equipment such as photo-copiers and cash registers, etc.

< previous | content | next >

< previous | content | next >

What is in VPortPro

VPortPro is an integrated and powerful 32-bit software tool consisting of the following functional parts:

VCOM-TCP Primary Client

The **VCOM-TCP Primary Client** is a software program designed to convert serial (RS232) data to the TCP/IP packets format and TCP/IP packets to serial data. While running on your PC, **VCOM-TCP Primary Client** can receive serial data from a serial port of your PC and then send them out to a TCP/IP network through a TCP/IP socket connection on your PC. **VCOM-TCP Primary Client** can also receive serial data from a TCP/IP network through a TCP/IP socket connection on your PC and send them to a serial port on your PC.

VCOM-TCP Primary Client supports two transport data stream protocols:

- Raw TCP connection (without sending configuration or status information from the PC to the serial device or server)
- Telnet (with the features provided by the COM Port Control protocol specified by IETF RFC-2217)

VCOM-TCP Primary Client functions as a TCP/IP client (for both the protocol types), i.e. it will initiate and establish connection with remote systems (by sending connection requests), and then communicate with connected TCP/IP servers. As a TCP/IP client, **VCOM-TCP Primary Client** allows to establish connection to a single TCP/IP server for each serial port. Normally any data received through the TCP/IP port that it connects to will be transmitted out the serial port and any data received through the serial port will be sent back to the TCP/IP server. This enables fully bi-directional communications. So, **VCOM-TCP Primary Client** redirects a local COM port to a TCP/IP port of a remote server and vice versa (using a local TCP/IP socket connection).

Also with **VCOM-TCP Primary Client**, you can create "virtual serial ports". A virtual serial port (VSP) appears to application software as a "standard" MS Window's communications port (virtually identical to an RS-232 port). Data may be written to VSP by applications, and data may be read from VSP as though it were a standard COM port. You can configure **VCOM-TCP Primary Client** to map server's TCP/IP ports onto created virtual serial ports. This allows MS Windows applications to generate or consume data to and from VSP as though the data were being processed by physical hardware.

You can use the virtual COM port driver on multi-processor and Hyper-Threading machines too.

While you are using **VCOM-TCP Primary Client**, you can:

- Configure COM-TCP/IP connection channels (i.e., make COM and TCP/IP mapping and settings for a connection channel)
- Manage the connection channels list (i.e., Add/Delete connection channels)
- Watch connection channels activity (i.e., status)
- Monitor data logging for a selected connection channel from the list
- Trace the **VCOM-TCP Primary Client** (driver) session using a variety of trace flags

Currently, **VCOM-TCP Primary Client** supports only its own virtual serial ports (i.e. only those it creates by itself). **VCOM-TCP Primary Client** does not support physical serial ports and virtual serial ports created by software tools other than VPortPro.

Currently, **VCOM-TCP Primary Client** cannot redirect a serial port to more than one remote TCP/IP port, and a remote TCP/IP port to more than one serial port.

COM Port Redirector

The **COM Port Redirector** is a software program designed to convert serial (RS232) data to the TCP/IP packets format and TCP/IP packets to serial data. While running on your PC, **COM Port Redirector** can receive serial data from a serial port of your PC and then send them out to a TCP/IP network through a TCP/IP socket connection on your PC. **COM Port Redirector** can also receive serial data from a TCP/IP network through a TCP/IP socket connection on your PC and send them to a serial port on your PC.

COM Port Redirector supports two transport data stream protocols:

- Raw TCP connection (without sending configuration or status information from the PC to the serial device or server)
- Telnet (with the features provided by the COM Port Control protocol specified by IETF RFC-2217)

COM Port Redirector can be configured to function as a TCP/IP server and as a TCP/IP client for both the protocol types.

As a TCP/IP client, **COM Port Redirector** will initiate and establish connection with remote systems (by sending connection requests), and then communicate with connected TCP/IP servers. As a TCP/IP client, **COM Port Redirector** allows to establish connection to a single TCP/IP server for each serial port. Normally any data received through the TCP/IP port that it connects to will be transmitted out the serial port and any data received through the serial port will be sent back to the TCP/IP server. This enables fully bi-directional communications. So, **COM Port Redirector** redirects a local COM port to a TCP/IP port of a remote server and vice versa (using a local TCP/IP socket connection).

As a TCP/IP server, **COM Port Redirector** will wait for remote TCP/IP clients' connection requests to TCP/IP ports (you specified as listening on your PC) to establish connection with remote systems, and then to communicate with connected TCP/IP clients. As a TCP/IP server, **COM Port Redirector** allows to accept only a single client connection to a TCP/IP port. Incoming data received from a TCP/IP client is transmitted out a serial port as it is received. Incoming serial data is sent to a connected TCP/IP client. This enables fully bi-directional communications. So, **COM Port Redirector** maps and redirects COM ports to a TCP/IP ports and vice versa.

Also with **COM Port Redirector**, you can create "virtual serial ports". A virtual serial port (VSP) appears to application software as a "standard" MS Window's communications port (virtually identical to an RS-232 port). Data may be written to VSP by applications, and data may be read from VSP as though it were a standard COM port. You can configure **COM Port Redirector** to map TCP/IP ports onto virtual serial ports as well as to physical COM ports. This allows MS Windows applications to generate or consume data to and from VSP as though the data were being processed by physical hardware.

You can use the virtual COM port driver on multi-processor and Hyper-Threading machines too.

While you are using **COM Port Redirector**, you can:

- Configure COM-TCP/IP connection channels (i.e., make COM and TCP/IP mapping and settings for a connection channel)
- Manage the connection channels list (i.e., Add/Delete connection channels)
- Watch connection channels activity (i.e., status)
- Monitor data logging for a selected connection channel from the list
- Trace the **COM Port Redirector** (driver) session using a variety of trace flags
- Connect/Disconnect connection channels (for the Client mode)

Currently, **COM Port Redirector** supports physical serial ports and virtual serial ports as well. Virtual serial ports can be created by COM Port Redirector itself or by software tools other than COM Port Redirector, for example, by VCOM-TCP Primary Client (COM Port Redirector will treat them as physical serial ports).

Currently, **COM Port Redirector** cannot redirect a serial port to more than one remote TCP/IP port, and a remote TCP/IP port to more than one serial port.

Also a single virtual serial port cannot simultaneously be used for both Client and Server modes.

COM Port Redirector Service

COM Port Redirector Service is a software program that designed to be used as a standard MS Windows service. The functionality and settings of Com Port Redirector Service are just the same as of Com Port Redirector itself. Additionally, you can control the behavior of Com Port Redirector

Service as a MS Windows service.

ComSetup

The configuration utility, ComSetup, allows you to make communication settings relating to the networking aspects of VPortPro that operate with the TCP/IP transport interface.

2. Introducing to VPortPro

< previous | content | next >

< previous | content | next >

Typical Usage of VPortPro Software

You can use the VPortPro's tools to solve the following typical tasks.

1. "Connecting" a serial device on one PC to a serial device on another PC

In this case (with both machines being accessible across a TCP/IP network and having VPortPro installed),

1. On the first PC:

- Configure **COM Port Redirector** for the server mode
- Map a COM port (to which your serial device is attached) to a local TCP/IP port
- Start up **COM Port Redirector**

2. On the second PC:

- Configure **COM Port Redirector** for the client mode
- Map a COM port (to which your serial device is attached) to the server's TCP/IP address and TCP/IP port (i.e., of the first PC)
- Start up **COM Port Redirector** and use Connect

After that, the two serial devices could "talk" with each other by passing their serial data across the TCP/IP network.

2. "Connecting" a serial device on one PC with an application running on another PC

In this case (with both machines being accessible across a TCP/IP network and having VPortPro installed),

1. On the first PC:

- Configure **COM Port Redirector** for the server mode

- Map a COM port (to which your serial device is attached) to a local TCP/IP port
- Start up **COM Port Redirector**

2. On the second PC:

- Configure **COM Port Redirector** for the client mode (or configure **VCOM-TCP Primary Client**)
- Configure your application to use a virtual serial port
- Map the virtual serial port (to which your application is "attached") to the server's TCP/IP address and TCP/IP port (i.e., of the first PC)
- Start up **COM Port Redirector** (or **VCOM-TCP Primary Client**) and use Connect

After that, the serial device and the application could communicate with each other by passing their serial data across the TCP/IP network. For example, the application program could remotely control modems, electronic monitoring equipment, testing lab equipment, data acquisition devices, etc.

3. "Connecting" an application running on one PC with an application running on another PC

In this case (with both machines being accessible across a TCP/IP network and having VPortPro installed),

1. On the first PC:

- Configure **COM Port Redirector** for the server mode
- Configure your application to use a virtual serial port
- Map the virtual serial port (to which your application is "attached") to a local TCP/IP port
- Start up **COM Port Redirector**

2. On the second PC:

- Configure **COM Port Redirector** for the client mode (or configure **VCOM-TCP Primary Client**)
- Configure your application to use a virtual serial port
- Map the virtual serial port (to which your application is "attached") to the server's TCP/IP address and TCP/IP port (i.e., of the first PC)
- Start up **COM Port Redirector** (or **VCOM-TCP Primary Client**) and use Connect

After that, the applications could "talk" with each other by passing their serial data across the TCP/IP network. You could use a terminal emulation program such as HyperTerminal to verify connectivity from **COM Port Redirector** to the serial device or server.

4. "Connecting" a serial device server to a serial device on your PC

In this case (with both your PC and the server being accessible across a TCP/IP network and having VPortPro installed on your PC),

- Configure **COM Port Redirector** for the client mode
- Map the COM port (to which your serial device is attached) to your serial device server's TCP/IP address and TCP/IP port
- Start up **COM Port Redirector** and use Connect

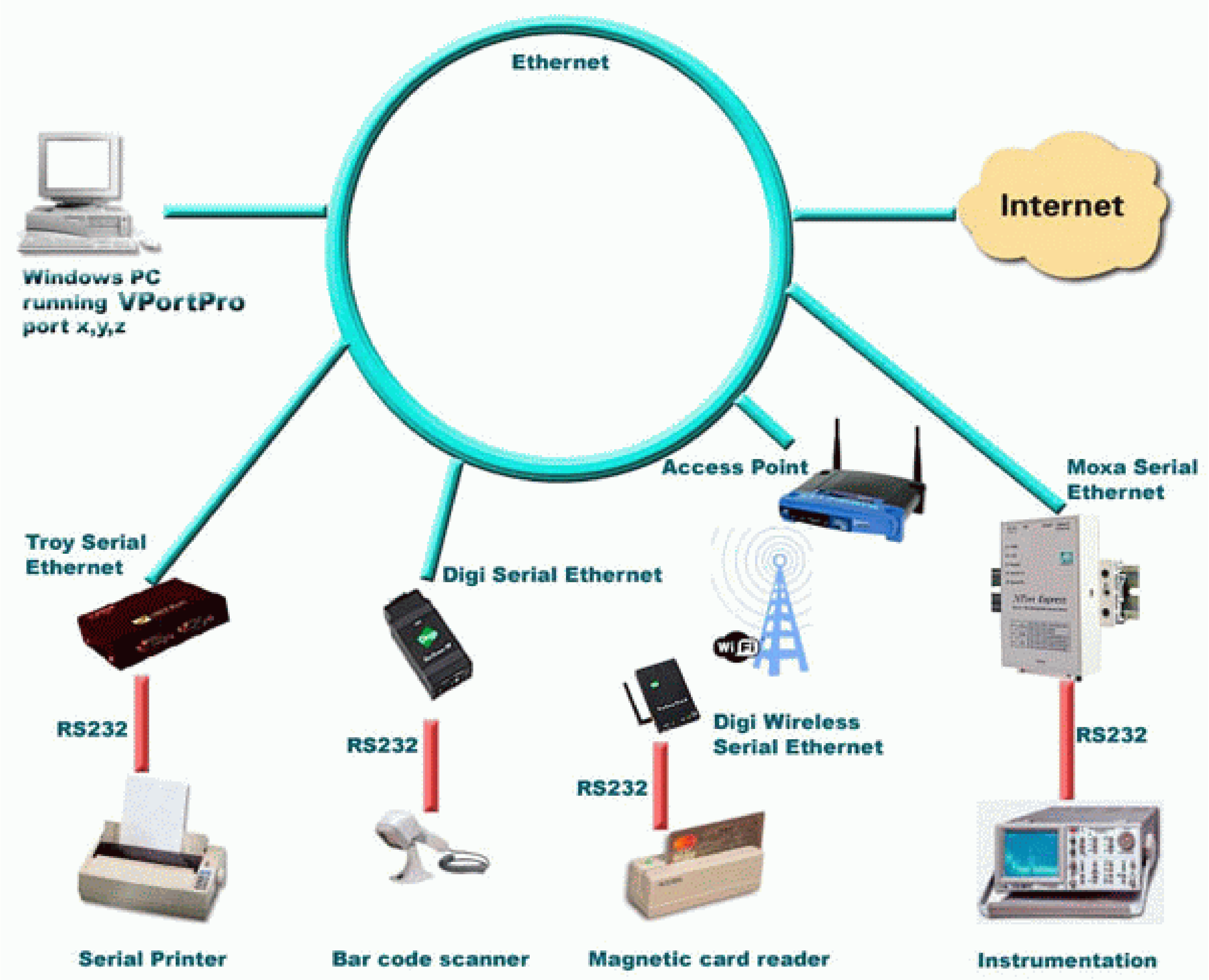
After that, the serial device and serial server could "talk" with each other by passing their serial data across the TCP/IP network.

5. "Connecting" a serial device server with an application running on your PC

In this case (with both your PC and the server being accessible across a TCP/IP network and having VPortPro installed on your PC),

- Configure **COM Port Redirector** for the client mode (or configure **VCOM-TCP Primary Client**)
- Configure your application to use a virtual serial port
- Map the virtual serial port (to which your application is "attached") to your serial device server's TCP/IP address and TCP/IP port
- Start up **COM Port Redirector** (or **VCOM-TCP Primary Client**) and use Connect

After that, the application and serial server could communicate with each other by passing their serial data across the TCP/IP network. For example, the application program could access and control serial devices connected to the serial server from any computer connected to the network. The application could set up and manage a lot of serial devices for remote monitoring system, attendance/entrance control systems, and many others making it possible to view images from the security cameras and check data from the sensors.



On the figure above, several serial servers are connected to the network. A serial device is attached to each serial server. Your PC is also connected to the network.

Configure VPortPro on your PC for the client mode. Create a virtual serial port and map it to the TCP/IP address and TCP/IP port of a server. You can do this for each serial device. Configure your application(s) to use these virtual serial ports. After that, the application is ready to communicate with the serial devices.

< previous | content | next >

< previous | content | next >

3. The VPortPro Requirements

Your computer system must meet the following hardware, software, host and network requirements for you to install and use VPortPro.

PC Hardware & Software Requirements

- A standard 32-bit IBM PC (e.g., i486, or Pentium) or 100% compatible
- 4 Mbytes RAM
- Color graphics controller supporting SVGA video modes
- Mouse Unit compatible with Microsoft Windows
- Network adapter for Ethernet or Serial port connection
- 9 Mbytes free hard disk space.

In addition to the above requirements, you need one of the following operating systems:

- MS Windows 2K/2K3/XP/XP-x64
- TCP/IP facility with Windows Sockets Interface.

Host Requirements

- TCP/IP protocols over Ethernet or Serial port connection

< previous | content | next >

< previous | content | next >

4. Installing VPortPro

This chapter describes how to install the VPortPro software. The chapter assumes that you have one of the Microsoft Windows 2K/2K3/XP/XP-x64 operating system installed as described in the corresponding user's guide for the product.

This chapter and the rest of the manual refer to the following two installation directories whose names you should specify at the installation steps:

- the home (or destination) directory (in which you install the package components files for all users)
- the configuration files directory (in which you store your particular configuration files, e.g. ini-files).

If you install VPortPro in a directory different from the default, simply supply your directory name when appropriate directories are requested.

The installation of the package is carried out by running the Setup program.

< previous | content | next >

< previous | content | next >

Running Setup

The software product comes normally as a self-extracting archive file that contains the package installation files.

The installation procedure consists of two steps:

1. Extracting the package installation files into a temporary distributive directory
2. Running Setup from the distributive directory.

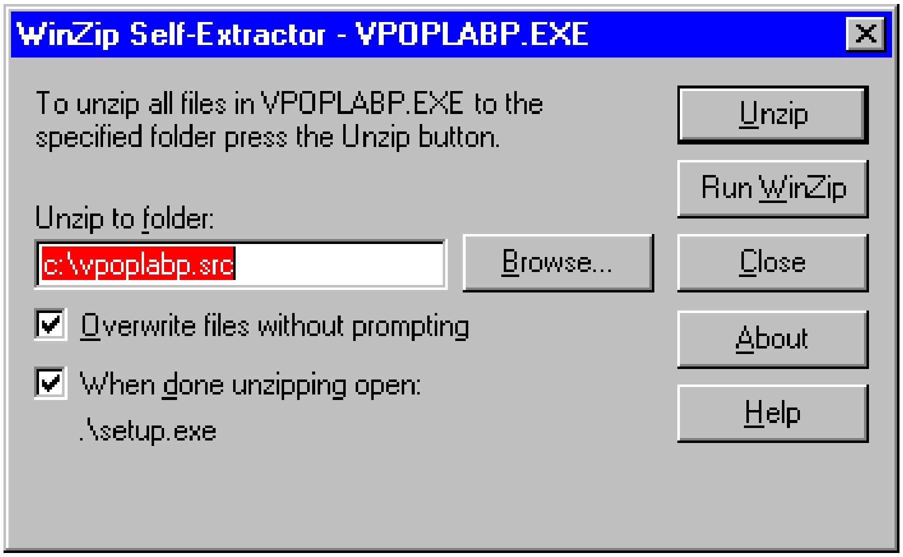
You can run Setup manually in the default mode or use the "silent" installation mode of Setup to simplify multiple secondary installations.

Extracting Package Installation Files

This section describes how to extract the package installation files from the self-extracting file (compressed and created by WinZip).

To extract the package installation files, you should do the following:

1. Download the self-extracting archive file, **vpoplabp.exe**, to your hard disk
2. Execute **vpoplabp.exe** and select **OK**. The dialog box will appear:


WinZip Self-Extractor - VPOPLABP.EXE

To unzip all files in VPOPLABP.EXE to the specified folder press the Unzip button.

Unzip to folder: c:\vpoplabp.src

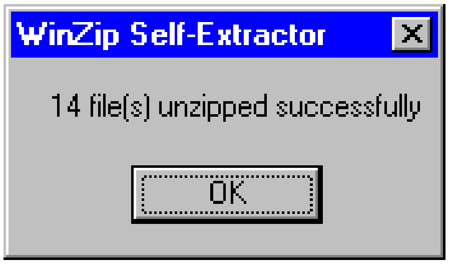
☑ Overwrite files without prompting

☑ When done unzipping open: .\setup.exe

You can enter a name for a temporary distributive directory in the **Unzip to folder** edit field or use the **Browse** button to select it. By default, the installation files will be extracted to distributive directory **c:\vpoplabp.src**.

Later (after successful installation), you can remove the temporary distributive directory or use it to start Setup for multiple secondary installations.

3. Choose **Unzip** to start extracting the files and then installing the package automatically. The **vpoplabp.exe** archive file will be uncompressed and the installation files will be placed in the specified distributive directory.
4. After extracting the package installation files, choose **OK** in the following box:


WinZip Self-Extractor

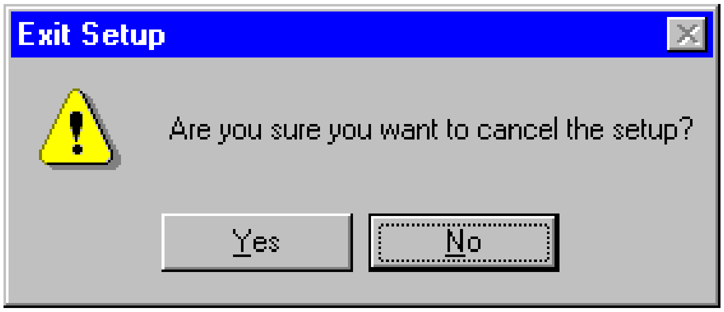
14 file(s) unzipped successfully

The installation procedure (i.e. running Setup) will start automatically from the temporary distributive directory if you enable the **When done unzipping open: .\setup.exe** check box.

Running Setup

As soon as you start the installation process, you will see a number of dialog boxes with instructions for each installation step. These boxes have three buttons. The **Cancel** button quits the installation process. The **Back** button returns you to the previous step. When you press the **Next** button, the Setup program proceeds to the next installation step.

At any step of installation, you can use the **Cancel** button. The **Exit Setup** window appears.



Exit Setup

Are you sure you want to cancel the setup?

You can confirm exiting or choose to continue installing.

For the first installation of the package, the procedure steps are as follows (with the dialog's names):

- **Welcome**

At this step, it is strongly recommended that you exit all MS Windows programs before running Setup.

- **Software License Agreement**

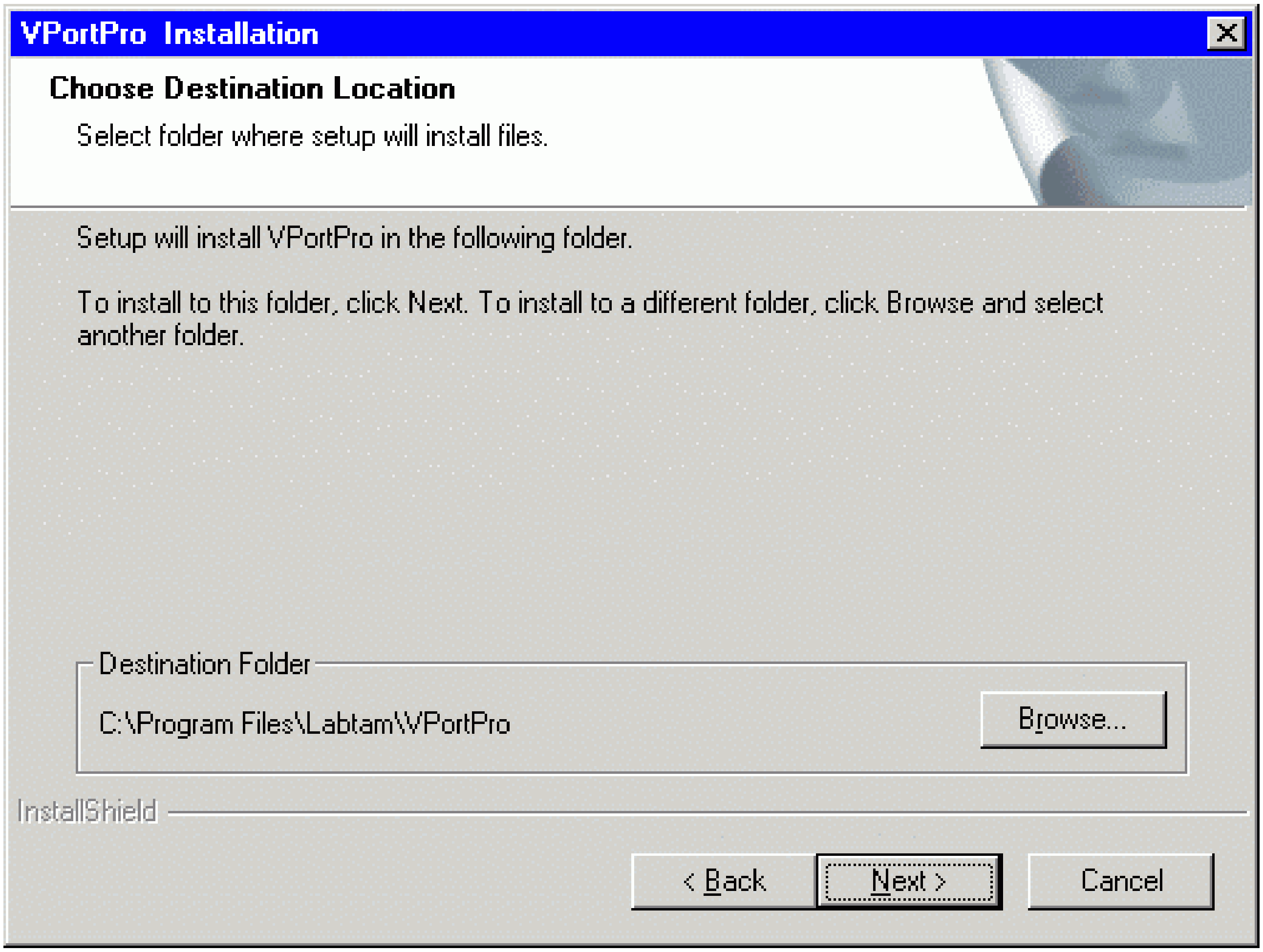
At this step, you must choose whether you accept all the terms of the Software License Agreement shown within the window or not before running Setup.

- **User Information**

In this box, you should enter the Person name, the Company name, and the product serial number (for registration purposes).

- **Choose Destination Location**

In this box, you should specify the folder where the package will be installed in (i.e. the home directory for the package). You can use the default folder name, enter your destination folder name, or select another folder using the **Browse** button.



VPortPro Installation

Choose Destination Location

Select folder where setup will install files.

Setup will install VPortPro in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

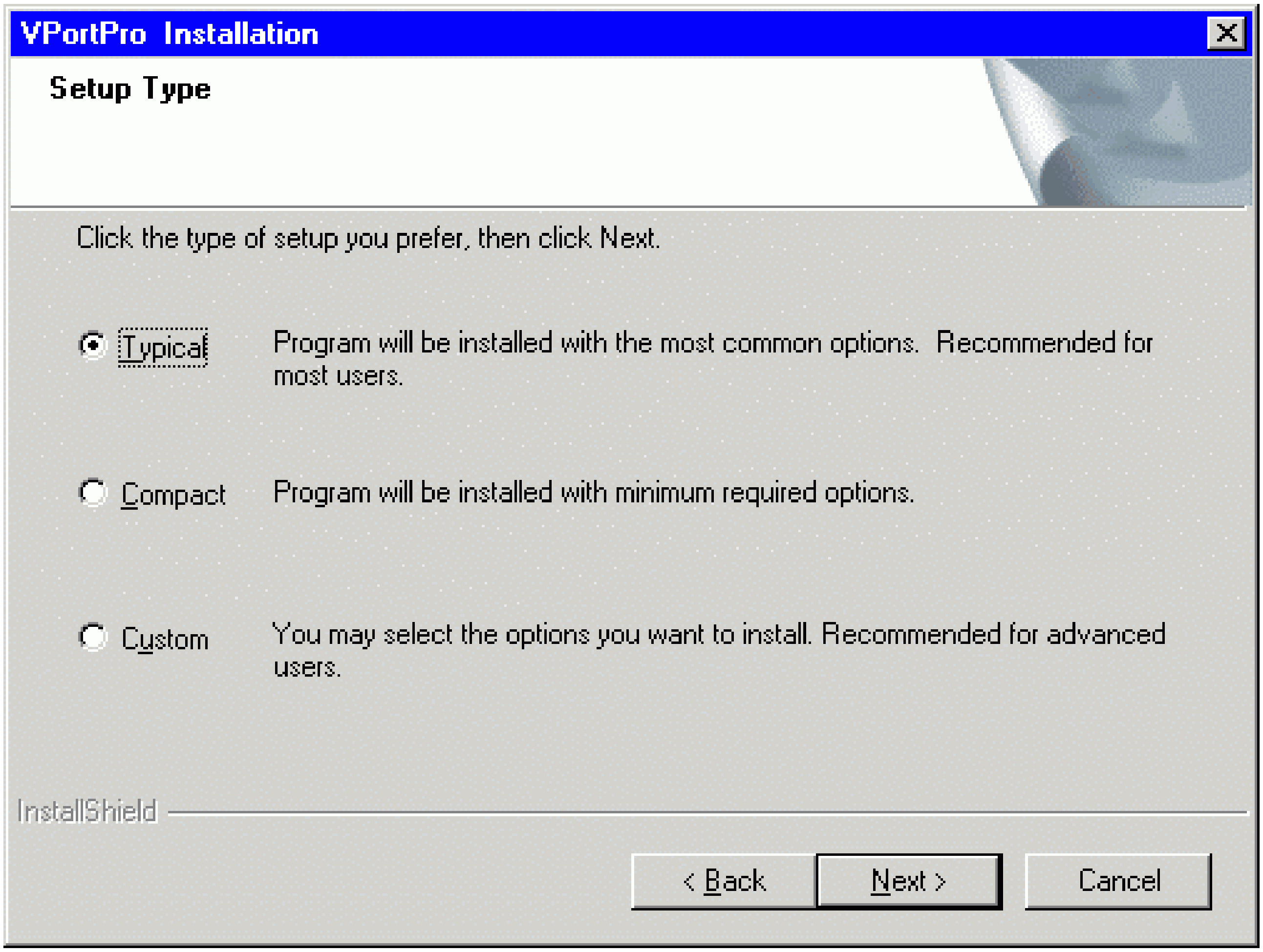
Destination Folder

C:\Program Files\Labtam\VPortPro

To install to the specified folder, click **Next**. If the folder does not exist, Setup will create it. If Setup detects the package in the directory you specified, then it will prompt you to upgrade the package. (See section **Upgrading VPortPro** below.)

- **Setup Type**

At this step, you can choose components of the package you want Setup to install.



VPortPro Installation

Setup Type

Click the type of setup you prefer, then click Next.

● Typical Program will be installed with the most common options. Recommended for most users.

○ Compact Program will be installed with minimum required options.

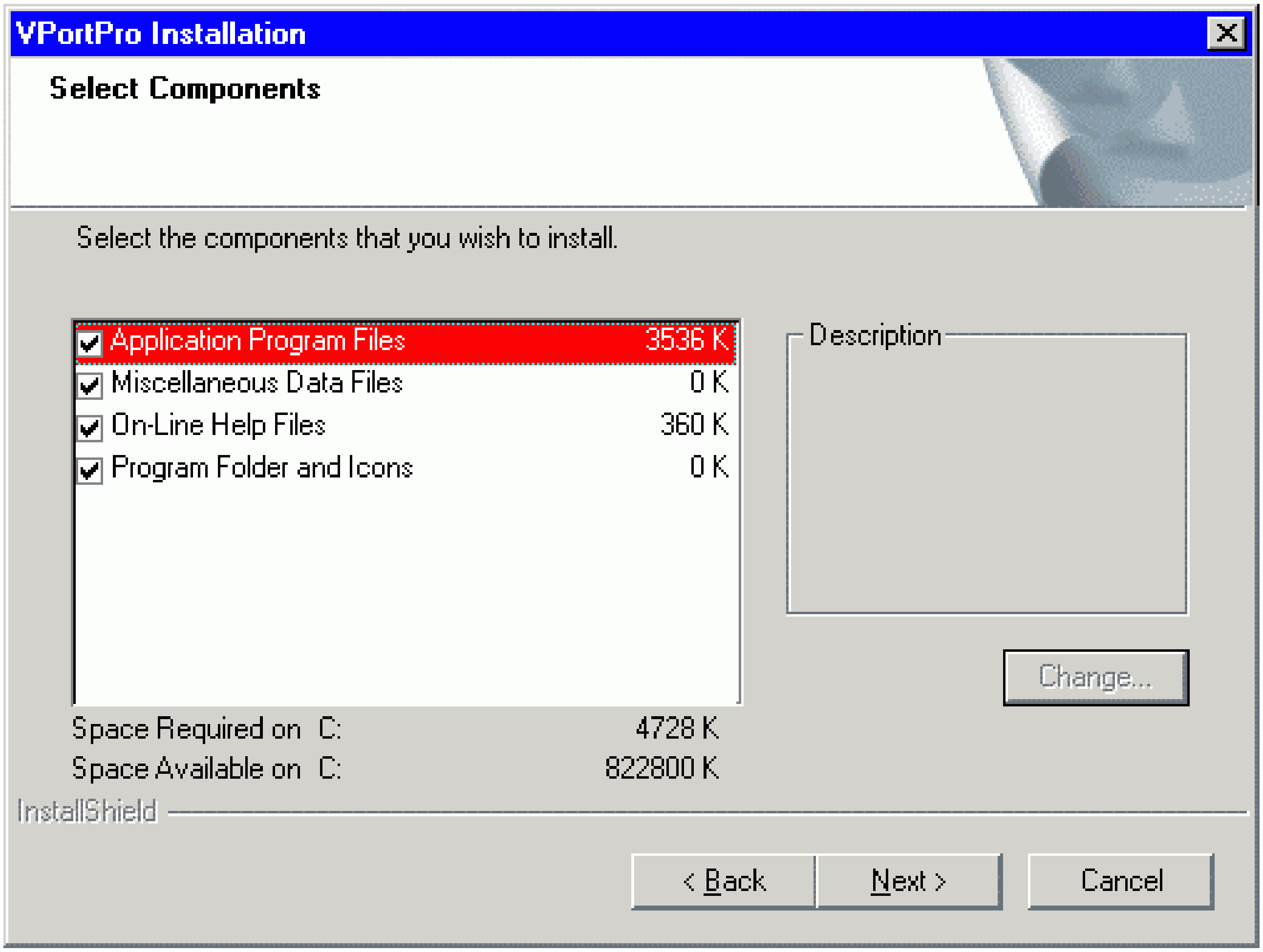
○ Custom You may select the options you want to install. Recommended for advanced users.

There are three Setup types: **Typical**, **Compact**, and **Custom**.

For the **Typical** type, Setup installs all the components of the package (the most common options).

For the **Compact** type, Setup installs the package without On-Line Help Files (the minimum required options).

For the **Custom** type, the **Select Components** dialog box appears on your display. Choose components that you wish to install (clicking on the checkmark positions).



VPortPro Installation

Select Components

Select the components that you wish to install.

☑ Application Program Files	3536 K
☑ Miscellaneous Data Files	0 K
☑ On-Line Help Files	360 K
☑ Program Folder and Icons	0 K

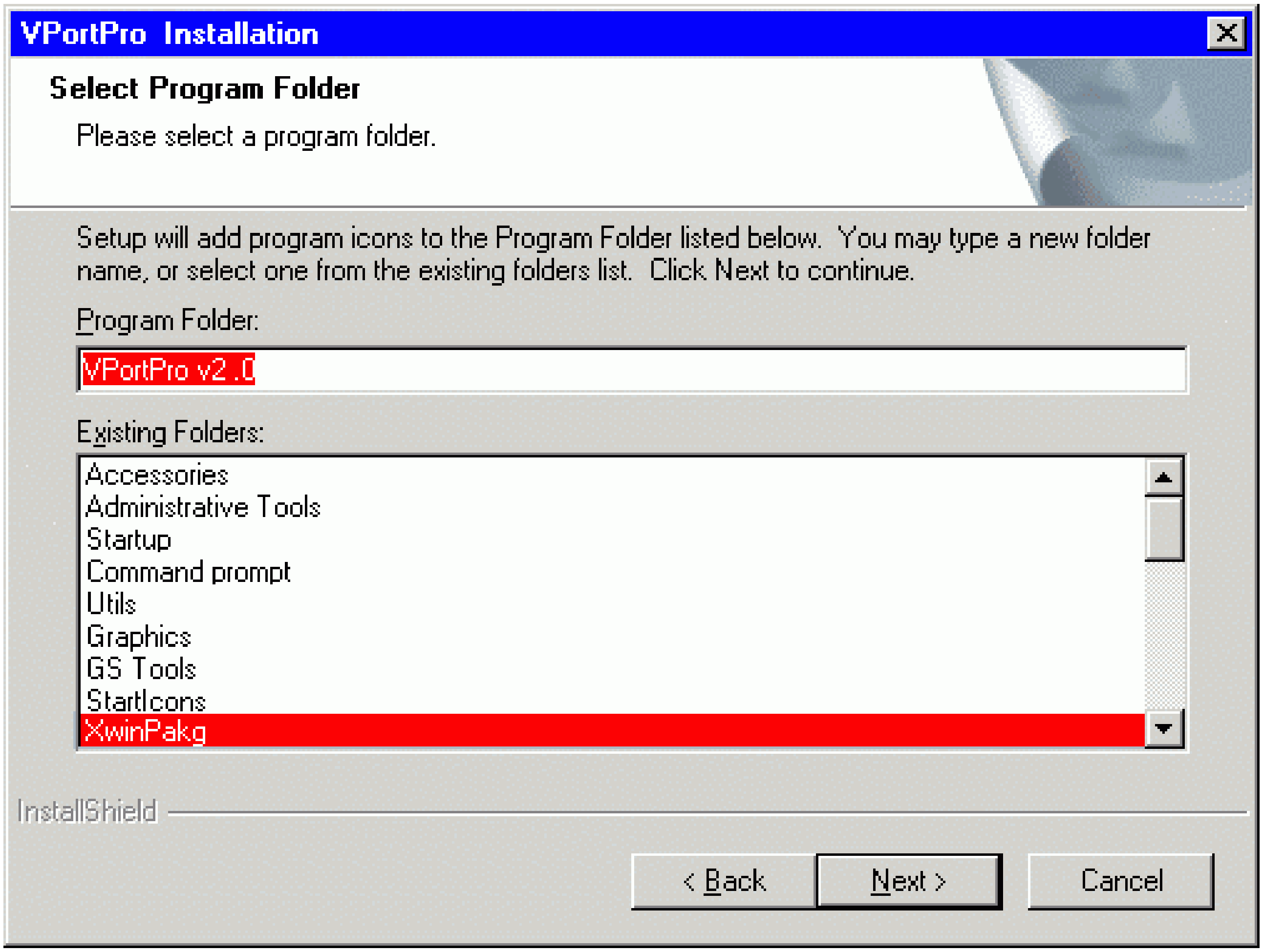
Space Required on C: 4728 K

Space Available on C: 822800 K

At the bottom of the window, you can see how much **Space Required** will be used to install the components you choose and **Space Available** on your disk.

- **Select Program Folder**

At this step, you should specify the Program Folder to contain shortcuts for the components being installed. In the **Select Program Folder** dialog box, you can use the default folder name, enter a new folder name, or select another folder from within the **Existing Folders** list.



VPortPro Installation

Select Program Folder

Please select a program folder.

Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing folders list. Click Next to continue.

Program Folder:

VPortPro v2 .0

Existing Folders:

- Accessories
- Administrative Tools
- Startup
- Command prompt
- Utils
- Graphics
- GS Tools
- StartIcons
- XwinPakg

Setup will add program icons to the Program Folder specified.

- **Setup**

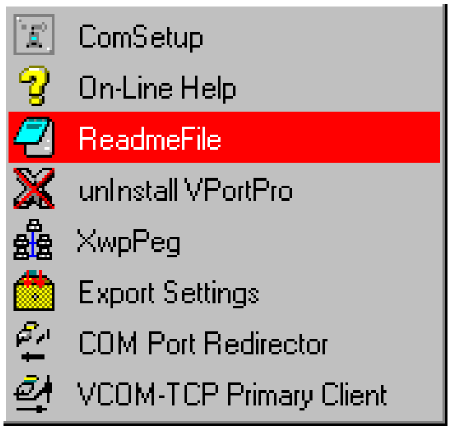
With the data specified, Setup begins to install the package. It shows how files of the components are being installed into the destination directory. Finally, Setup creates the VPortPro Program folder with shortcuts for the package components. It also adds the VPortPro item to the Programs menu.

- **Setup complete**

At this step, Setup informs you that the package is ready to run. In the **Setup complete** dialog box, click **Finish** to complete Setup.

The VPortPro Programs' Folder

You may run the installed programs by clicking program's icons from the VPortPro Programs' folder:



The **COM Port Redirector** item brings up the **COM Port Redirector** window where you can manage a connection channels list, make settings for virtual serial ports and TCP/IP ports, and control activity of a selected connection channel. (Refer to chapter **COM Port Redirector**.)

The **ComSetup** item brings up the **Communication Setup / Windows Sockets** dialog box where you can make communication settings relating to the networking aspects of VPortPro that operate with the TCP/IP transport interface. (Refer to section **Using ComSetup** in chapter **Configuring VPortPro**.)

The **Export Settings** item is used to save all the settings made on the COM-TCP/IP connection channel list in the **VCOM-TCP Primary Client** and **COM Port Redirector** windows into the **vppini.dat** file. (Refer to chapter **Export Settings**.)

The **On-Line Help** item displays this manual.

The **ReadmeFile** item displays the **Readme** file (placed in the installation folder) that contains the latest information and product notes about the VPortPro package.

The **Uninstall VPortPro** item is used to uninstall the VPortPro package from your computer. (Refer to section **Running UNINSTALL** in chapter **Installing VPortPro**.)

The **VCOM-TCP Primary Client** item brings up the **VCOM-TCP Primary Client** window where you can manage a connection channels list, make settings for virtual serial ports and TCP/IP ports, and control activity of a selected connection channel in the **Client** mode. (Refer to chapter **VCOM-TCP Primary Client**.)

The **XwpPeg** item launches the XwpPeg utility that checks and displays the state of the package license accounts. (Refer to subsection **The XwpPeg Utility** in section **Using ComSetup** in chapter **Configuring VPortPro**.)

The Silent Installation Mode

The "silent" installation mode of Setup may be used for multiple secondary installations on different computers working with similar package installation environment (i.e. drive letters, installation directory name, presence of the package, etc).

The "silent" installation process consists of the following two steps:

1. Normal first installing of the package with creating a script file
2. Using the script file for multiple secondary installations.

1) To install the package with creating the script file, **setup.iss**, run Setup with the options:

```
setup -r -f1PATH\setup.iss
```

The **-f1** option enables you to specify an alternative file location and script file name. It is recommended to specify the absolute PATH for the option.

With these arguments, Setup performs normal installation of the package and creates the script file you specified in the command in the directory according to the specified PATH. The file contains data you specified for Setup to install the package. (See section **Running Setup** above.)

Setup performs secondary installations using the script file, so you have to specify no input data (you only watch "silently" how Setup works automatically).

2) To perform secondary installation, make sure that the script file created at the first step is located in the distributive directory (where the **setup.exe** file exists). If not, then copy the script file to the distributive directory, and then run Setup with the following command line:

```
setup -s -wauto
```

With this argument (and without the **-f1** option), Setup will install the package according to the **setup.iss** script file. Setup will search for the file in the distributive directory.

Also, you can use the following command line to perform secondary installation:

```
setup -s
```

This is the same as the "-wauto" option with the only difference: the Finish dialog message is suppressed in this case.

When running an InstallScript MSI or InstallScript installation in silent mode (i.e., using the **-s** option), the log file, **setup.log**, is by default created in the same directory and with the same name (except for the extension) as the response file. The **-f2** option enables you to specify an alternative file location and log file name. It is recommended to specify the absolute path for the option as in the following example:

```
.\PackInstall\Setup.exe -s -f1C:\PackInstall\Setup.iss -f2C:\PackInstall\Setup.log
```

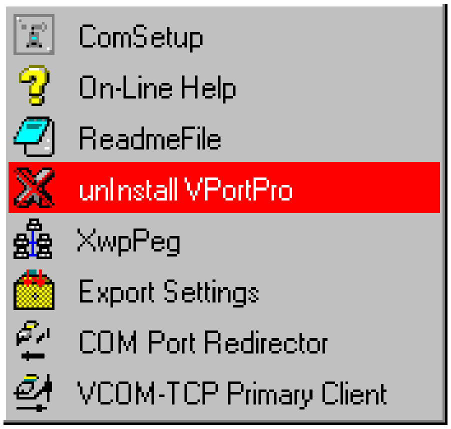
After Setup has finished (successfully or not) you can find the ASCII tracing file, **mkltrace.out**, in the distributive directory and look it through for error messages. Note that the "silent" installation may require interactive actions if Setup detects serious problems.

< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Running UNINSTALL

You can uninstall the package by clicking on the **Uninstall VPortPro** item in the VPortPro Programs' folder (i.e., from the Start/Programs/VPortPro menu):



The program will prompt you to confirm removing the package from your computer.

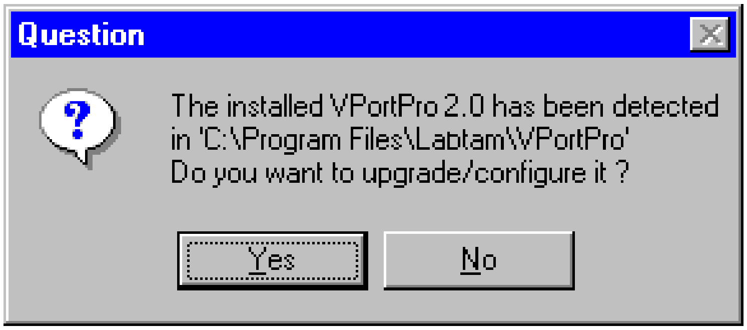
When **Uninstall VPortPro** completed, some elements might not be removed. You should manually remove items related to the application.

Note that all the virtual serial ports created with the virtual serial port driver of **VPortPro** will be removed from your system.

< previous | content | next >

Upgrading VPortPro

If you specify the home directory that contains the package installation files, the Setup program will detect it and prompt you to upgrade or configure it.

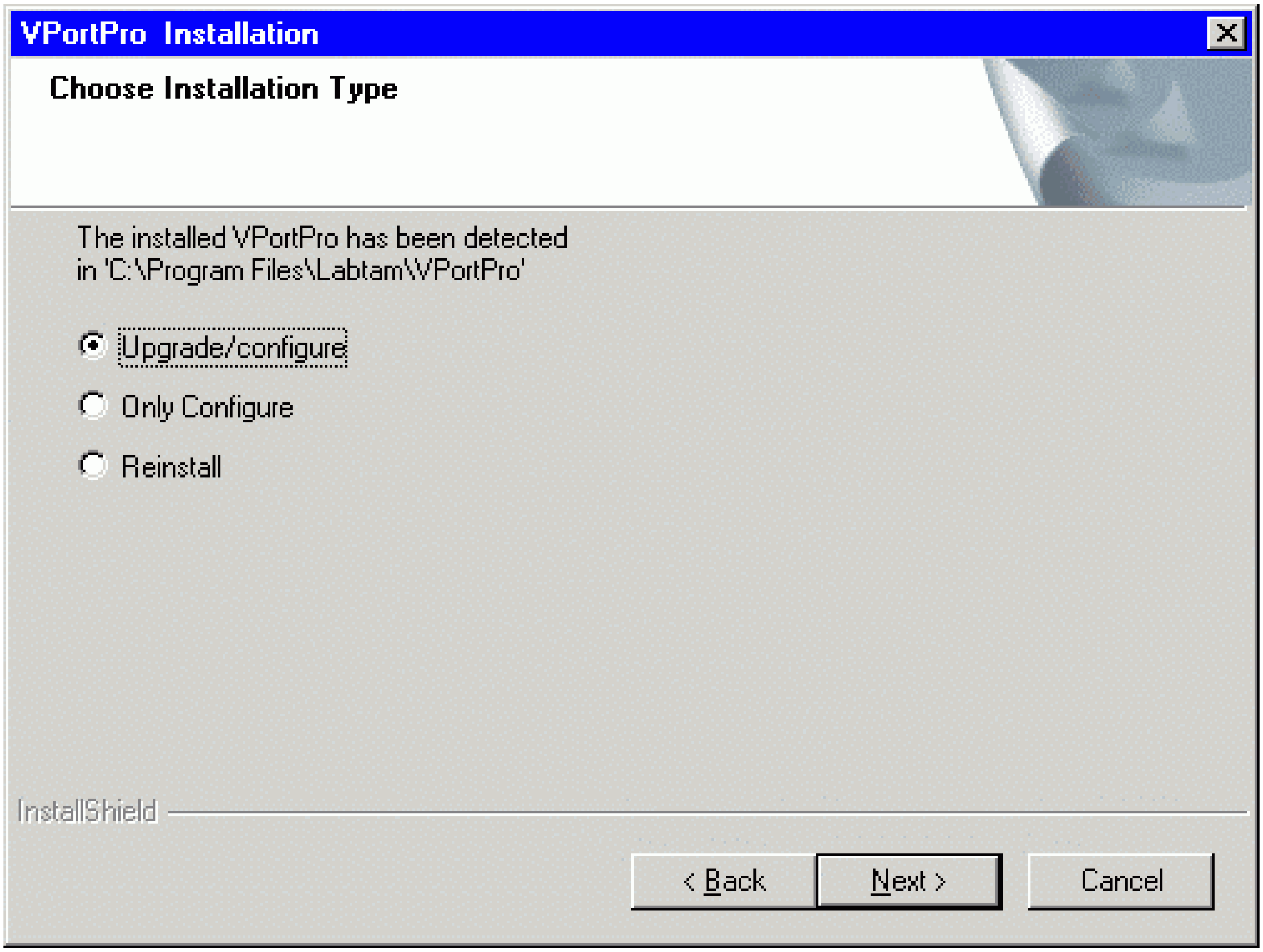

Question

The installed VPortPro 2.0 has been detected
in 'C:\Program Files\Labtam\VPortPro'
Do you want to upgrade/configure it ?

If you choose **YES**, i.e. to upgrade the installed package, Setup brings up the **Upgrade Type** window. You can choose **Typical**, **Compact**, or **Custom** installation type (like you do it at step **Setup Type** of the normal installation sequence). This allows you to upgrade the package or to reconfigure it without reinstalling binary programs.

If you choose **NO**, Setup returns you to the **Software License Agreement** step of the normal installation sequence. This allows you to completely reinstall the package.

If you specify the destination directory that exists, Setup brings up the **Choose Installation Type** window.


VPortPro Installation

Choose Installation Type

The installed VPortPro has been detected
in 'C:\Program Files\Labtam\VPortPro'

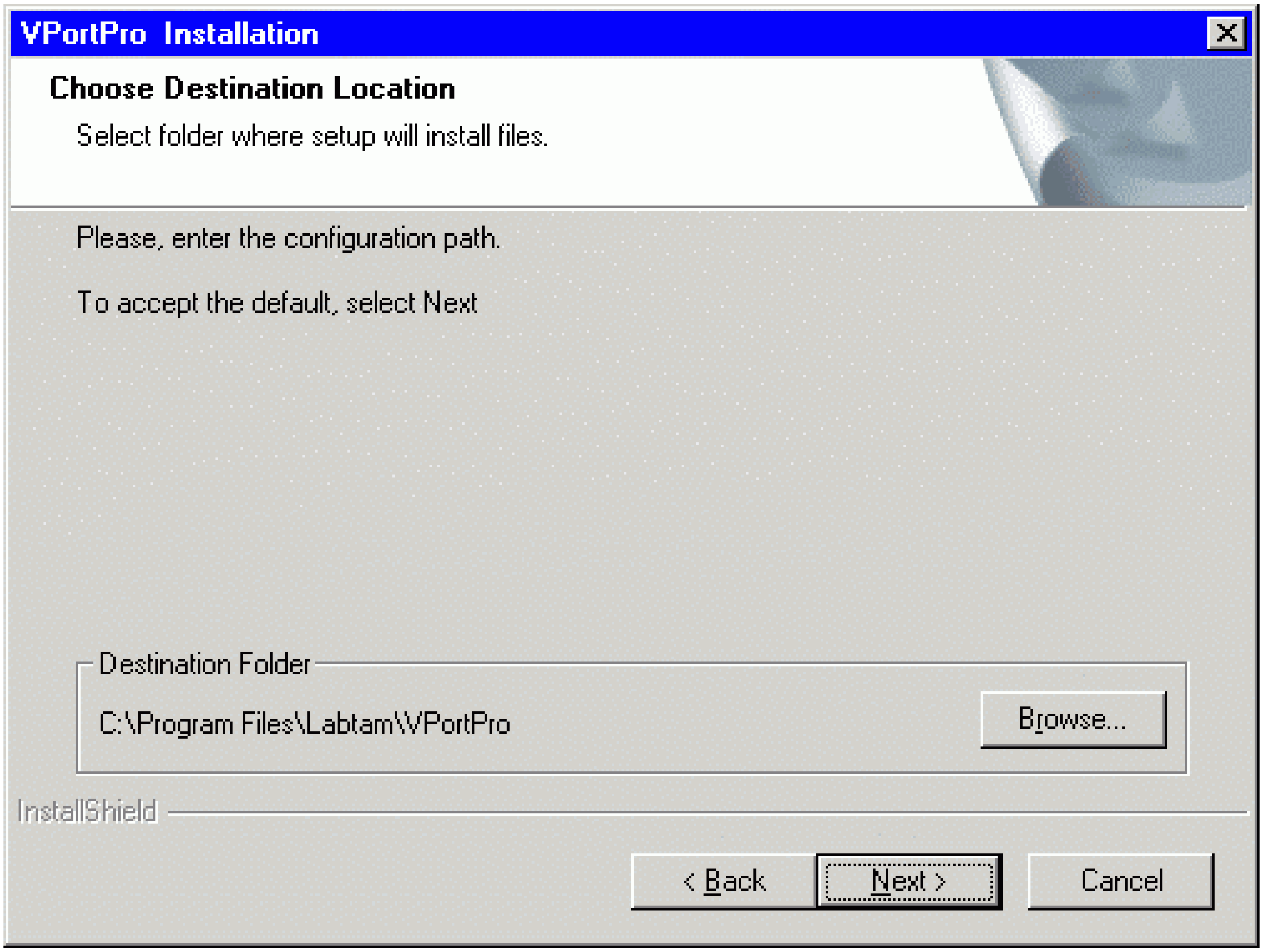
● Upgrade/configure

○ Only Configure

○ Reinstall

If you choose **Reinstall**, Setup will return you to the **Setup Type** step.

If you choose **Upgrade/configure**, Setup returns you to the **Upgrade Type** step. If you choose **Only Configure**, Setup goes to the **Select Components** step. Then, in both cases, the **Choose Configuration Location** window appears.



VPortPro Installation

Choose Destination Location

Select folder where setup will install files.

Please, enter the configuration path.

To accept the default, select Next

Destination Folder

C:\Program Files\Labtam\VPortPro

In this box, you should specify the configuration directory (i.e. configuration path) in which you store your particular configuration files (e.g. ini-files) and the configuration components you selected. Then, Setup prompts you to **Select Program Folder**.

Note that if you choose **Only Configure,** then Setup will make no changes in the home directory of VPortPro detected.

4. Installing VPortPro

< previous | content | next >

Multi-user Installation

Multi-user installation is intended for installing a single copy of VPortPro on a disk that will be shared by multiple users. VPortPro must be configured for each user on each PC it will be used on. Corresponding files that define user's local VPortPro configuration will be created in a specified directory (see **Choose Configuration Location**).

Multi-user installation is carried out as follows:

1. By running the Setup program, install the package on a disk that will be used for storing the shared copy (i.e. under the home directory of VPortPro).

The next step must be done by every user of the shared copy.

2. Run the Setup program. In the installation dialogs, you must specify the home directory of VPortPro for the shared copy (step **Choose Destination Location**), and a local directory where files defining a particular package configuration will be resident (step **Choose Configuration Location**).

Note that these secondary steps make no changes in the home directory of VPortPro created at the first step.

< previous | content | next >

5. Configuring VPortPro

This chapter describes how to configure the VPortPro package with the configuration utility. The utility allows you to set up VPortPro for your preferences, your host system and your PC.

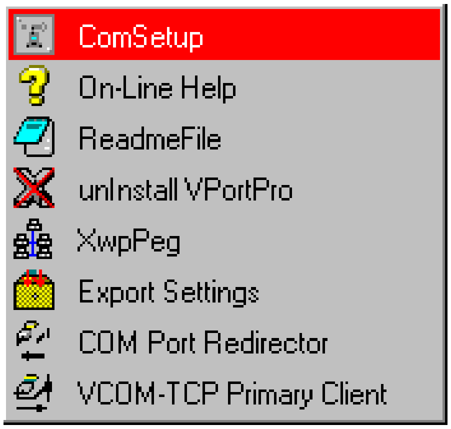
The configuration utility, ComSetup, included in VPortPro allows you to make communication settings relating to the networking aspects of VPortPro that operate with the TCP/IP transport interface.

< previous | content | next >

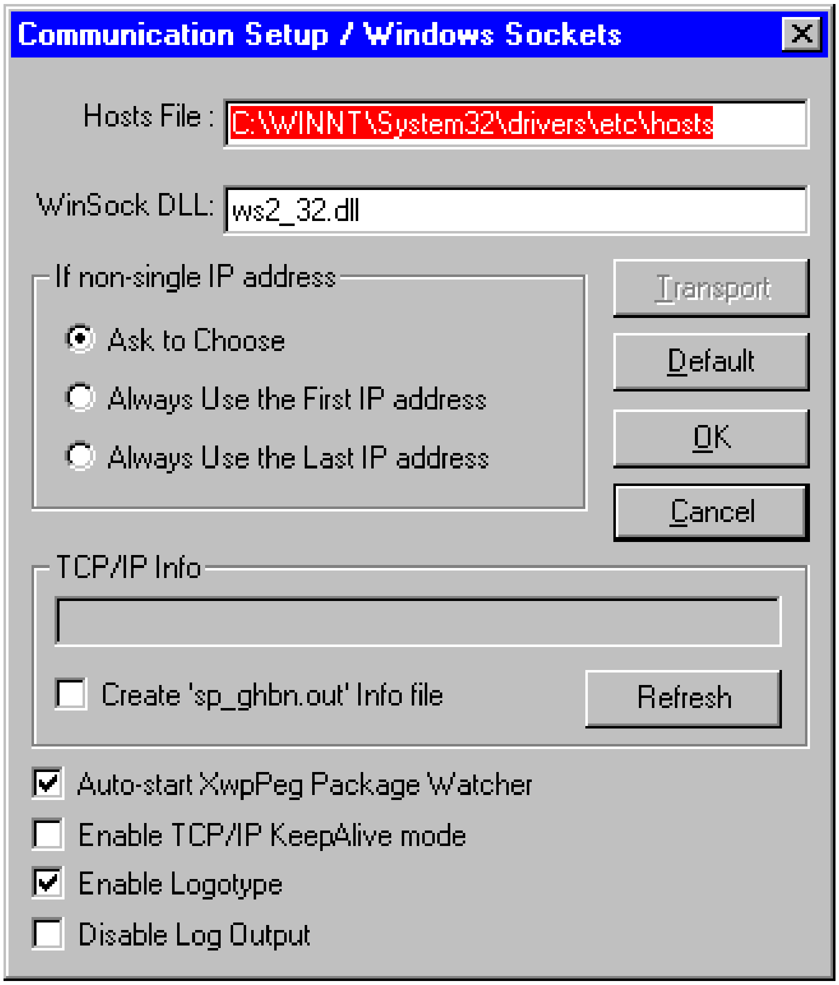
< **previous** | **content** | **next** >

Using ComSetup

You can start ComSetup by clicking on the **ComSetup** item in the VPortPro Programs' folder (i.e., from the Start/Programs/VPortPro menu):



The **Communication Setup / Windows Sockets** dialog box will appear on your display:



Hosts File : C:\WINNT\System32\drivers\etc\hosts

WinSock DLL: ws2_32.dll

If non-single IP address

● Ask to Choose
○ Always Use the First IP address
○ Always Use the Last IP address

TCP/IP Info

□ Create 'sp_ghbn.out' Info file

☑ Auto-start XwpPeg Package Watcher
□ Enable TCP/IP KeepAlive mode
☑ Enable Logotype
□ Disable Log Output

By clicking **OK**, any new settings you make to the **Communication Setup** dialog box will be saved in the **xwp.ini** file (by default) and the dialog will be closed. (See also section **Running ComSetup with command line parameters** below.)

You can cancel any changes you have made to the dialog box and close it by pressing **Cancel**.

There are some additional settings you can specify to customize ComSetup. They are described in section **The "[NETWORK]" Section of the ini-file** below.

The following input fields are available:

The Hosts File

This field is used to specify a location of the **hosts** file. You must enter the **hosts** file that is used by MS Windows (**C:\WinNT\System32\drivers\etc\hosts** for MS Windows NT4/2000/XP and **C:\Windows\hosts** for MS Windows 9x/ME).

The **hosts** file is a list of remote computers in the standard format (IP-address hostname aliases). The contents of the **hosts** file will be used by other programs of the package when you want to select a host to connect to from the host list box.

WinSock DLL

This field specifies the DLL that provides Windows Sockets Interface to existing TCP/IP stack. By default, **wsock32.dll** of the Microsoft Windows' TCP/IP will be used. You can specify to use any other TCP/IP stack by entering its 32-bit Windows Sockets Interface DLL.

Auto-start XwpPeg Package Watcher

With this check box enabled, the XwpPeg utility is started automatically with starting any package's application. Otherwise, you can launch XwpPeg manually.

Enable TCP/IP KeepAlive mode

This check box specifies the package's applications to use the TCP/IP feature, KeepAlive, when communicating with remote computers over your network. When enabled, this prevents your connection from interrupting by a remote computer when your PC does not send messages to it for a long time.

Enable Logotype

This check box toggles displaying the Logotype image each time the package's application starts up.

Disable Log Output

When selected, this check box prevents any program of the package from writing log information to the ".out" and ".ini" files.

Transport

This button lets you choose the network transport.

Default

This button will initialize all these parameters to their default values.

The If non-single IP address Box

If your PC has more than one IP address (i.e. 'multi-home' PC – with non-single TCP/IP stack, e.g. for Ethernet + modem), then you should specify a mode for choosing one of them.

Always Use the First IP address
Always Use the Last IP address

These modes allow VPortPro's programs to automatically choose the local IP address.

Ask to Choose

You can set up this mode to specify that you will choose the address in the dialog box brought up by the programs. The default mode is **Ask to Choose**.

The TCP/IP Info Box

Refresh

When you click on this button, ComSetup will search for available TCP/IP information and, if found, display in the info field the IP address and name of your PC according to mode settings.

Create 'sp_ghbn.out' Info file

If this check box is enabled, then all information found will be stored in the file. This allows you to check accessibility and obtain description of the TCP/IP stack used.

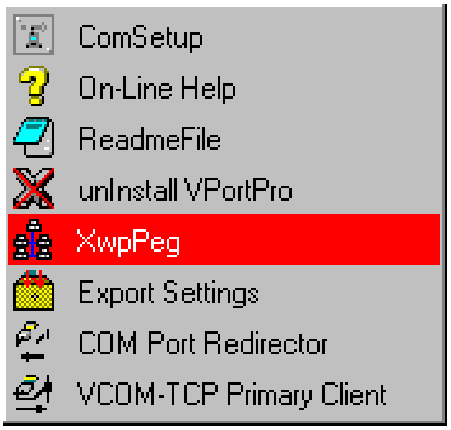
The XwpPeg Utility

XwpPeg is a program designed to check the state of the package license accounts. XwpPeg informs users about exceeding the allocated license accounts over the specified limit. If all users of the same package have XwpPeg running simultaneously with running any package's application (and terminate after completion of the last one) then all those users will have true information about the number of concurrent package users who use package license accounts (i.e. the allocated accounts).

You can launch XwpPeg either automatically or manually. With the **Auto-start XwpPeg Package Watcher** check box enabled, XwpPeg is started automatically with starting any package's application.

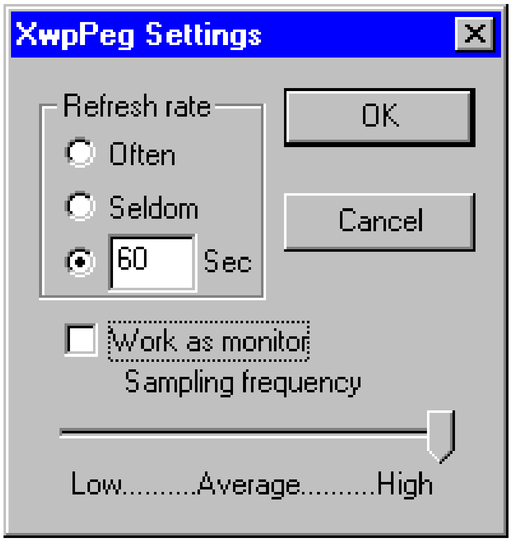
Users can independently launch XwpPeg (before starting any package's application) to check/allocate one license account in advance. If the user starts XwpPeg manually then he/she must also manually terminate it to free the package license account allocated. Since XwpPeg is only informational, users can freely terminate it at any time; however they cannot automatically check the current number of concurrent users of the same package in that case.

You can start XwpPeg by clicking on the **XwpPeg** item in the VPortPro Programs' folder (i.e., from the Start/Programs/VPortPro menu):



If you do not want to display the XwpPeg window, you should set to zero the value of the WorkMonitor variable in the "[XWPWATCH]" section of your **xwp.ini** file (e.g., **WorkMonitor=0**). To display the window, set the value to unit.

If you click on the **Settings** button, the following dialog box appears:


XwpPeg Settings

Refresh rate
○ Often
○ Seldom
● 60 Sec

□ Work as monitor

Sampling frequency

When you have made desirable settings, press **OK**.

The TCP/IP Retransmission Timeout Parameters

TCP starts a retransmission timer when each outbound segment is handed down to IP. If no acknowledgment has been received for the data in a given segment before the timer expires, then the segment is retransmitted, up to the TcpMaxDataRetransmissions times. The default value for this parameter is 5.

When a TCP connection is established, the retransmission timer is initialized to three seconds; however, it is adjusted on the fly to match the characteristics of the connection using Smoothed Round Trip Time (SRTT) calculations (as described in RFC793). The timer for a given segment is doubled after each retransmission of that segment. Using this algorithm, TCP tunes itself to the normal delay of a connection. TCP connections over high-delay links will take much longer to time out than those over low-delay links.

By default, after the retransmission timer hits 240 seconds, it uses that value for retransmission of any segment that needs to be retransmitted. This can be a cause of long delays for a client to time out on a slow link.

MS Windows NT4/2000 provide a mechanism to control the initial retransmit time, and then the retransmit time is self-tuning. The following is based on the Microsoft Knowledge Base and Microsoft MSDN Library.

To change the initial retransmit timeout parameters, you can modify the following values in the following registry key:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters

Value Name: InitialRtt
Data Type: REG_DWORD
Valid Range: 0-65535 (decimal)
Default: 0xBB8 (3000 decimal)

The **InitialRtt** parameter controls the initial retransmission timeout used by TCP on each new connection. It applies to the connection request (SYN) and to the first data segment(s) sent on each connection. For example, the value data 5000 decimal sets the initial retransmit time to five seconds.

Value Name: TcpMaxDataRetransmissions
Data Type: REG_DWORD - Number
Valid Range: 0 - 0xFFFFFFFF
Default: 5

The **TcpMaxDataRetransmissions** parameter controls the number of times TCP retransmits an individual data segment (non connect segment) before aborting the connection. The retransmission timeout is doubled with each successive retransmission on a connection. It is reset when responses resume. The base timeout value is dynamically determined by the measured round-trip time on the connection.

Caution

The above text contains information about editing the registry. Before you edit the registry, make sure you understand how to restore it if a problem occurs. Using Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk.

For information about how to edit and restore the registry, view the "Changing Keys and Values" and "Restoring the Registry" Help topics in Registry Editor (in **Regedit.exe**) or the "Add and Delete Information in the Registry", "Edit Registry Data", and "Restoring a Registry Key" Help topics (in **Regedt32.exe**).

Note that you should back up the registry before you edit it. If you are running MS Windows NT/2000, you should also update your Emergency Repair Disk (ERD).

< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Running ComSetup with Command Line Parameters

The 'xini' parameter

You can launch ComSetup with the 'xini' command line parameter:

PATH\comsetup.exe -xini <IniFilePath>

where <IniFilePath> specifies a full path to a specific ini-file and PATH indicates your home directory of VPortPro.

This feature allows you to run several ComSetup sessions each with its own ini-file (i.e. settings).

In order to do so, you can create a new ComSetup shortcut (e.g., in the VPortPro Programs' folder) and fill in the **Target** field in Properties of it with the command. For example, the **Target** field in Properties of the ComSetup shortcut might contain the following line:

"PATH\comsetup.exe" -xini myxwp1.ini

By default, the field contains a call of ComSetup with no arguments, and the **xwp.ini** file will be used in this case.

To create your specific ini-file, you can copy the **xwp.ini** file and then change required parameters with the ComSetup utility by starting it with the command line parameter.

5. Configuring VPortPro < **previous** | **content** | **next** >

The "[NETWORK]" Section of the ini-file

The "[NETWORK]" section of the ini-file may have the following entry lines you can customize for particular needs and applications. These settings are absent in the **Communication Setup** dialog box.

KeepAlive=1

This entry line specifies the VPortPro's software to use the TCP/IP feature, KeepAlive, when communicating with remote computers over your network.

This prevents your connection from interrupting by a remote computer when your PC does not send messages to it for a long time.

MyIP=virtual_local_IP_address

This entry line provides a "virtual" local IP address that is sent to a remote host instead of the real value.

Note: This setting is necessary for some programs running on remote computers to communicate with your PC from outside proxy (e.g., for TTSSH and other proxy-like programs).

5. Configuring VPortPro < **previous** | **content** | **next** >

< previous | content | next >

6. VCOM-TCP Primary Client

This chapter describes how to start and use the **VCOM-TCP Primary Client** program supplied with VPortPro.

The **VCOM-TCP Primary Client** is a software program designed to convert serial (RS232) data to the TCP/IP packets format and TCP/IP packets to serial data. While running on your PC, **VCOM-TCP Primary Client** can receive serial data from a serial port of your PC and then send them out to a TCP/IP network through a TCP/IP socket connection on your PC. **VCOM-TCP Primary Client** can also receive serial data from a TCP/IP network through a TCP/IP socket connection on your PC and send them to a serial port on your PC.

You could connect a serial device to a COM port on your PC, run **VCOM-TCP Primary Client** and then connect to the device from any other PC on your network through a TCP/IP socket connection. This would allow you to use Telnet or any other TCP/IP communications software to read or write to the serial device directly from any PC located on the same network.

A terminal server is a device that connects a serial instrument to a network and provides an I/O path to the device through a TCP/IP port. You can configure **VCOM-TCP Primary Client** for several ports at a time and associate each serial port to a different remote TCP/IP port (e.g., of a serial server).

VCOM-TCP Primary Client supports two transport data stream protocols:

- Raw TCP connection (without sending configuration or status information from the PC to the serial device or server)
- Telnet (with the features provided by the COM Port Control protocol specified by IETF RFC-2217)

VCOM-TCP Primary Client functions as a TCP/IP client (for both the protocol types), i.e. it will initiate and establish connection with remote systems (by sending connection requests), and then communicate with connected TCP/IP servers. As a TCP/IP client, **VCOM-TCP Primary Client** allows to establish connection to a single TCP/IP server for each serial port. Normally any data received through the TCP/IP port that it connects to will be transmitted out the serial port and any data received through the serial port will be sent back to the TCP/IP server. This enables fully bi-directional communications. So, **VCOM-TCP Primary Client** redirects a local COM port to a TCP/IP port of a remote server and vice versa (using a local TCP/IP socket connection).

Also with **VCOM-TCP Primary Client**, you can create "virtual serial ports". A virtual serial port (VSP) appears to application software as a "standard" MS Window's communications port (virtually identical to an RS-232 port). Data may be written to VSP by applications, and data may be read from VSP as though it were a standard COM port. You can configure **VCOM-TCP Primary Client** to map server's TCP/IP ports onto created virtual serial ports. This allows MS Windows applications to generate or consume data to and from VSP as though the data were being processed by physical hardware.

You can use the virtual COM port driver on multi-processor and Hyper-Threading machines too.

While you are using **VCOM-TCP Primary Client**, you can:

- Configure COM-TCP/IP connection channels (i.e., make COM and TCP/IP mapping and settings for a connection channel)
- Manage the connection channels list (i.e., Add/Delete connection channels)
- Watch connection channels activity (i.e., status)
- Monitor data logging for a selected connection channel from the list
- Trace the **VCOM-TCP Primary Client** (driver) session using a variety of trace flags

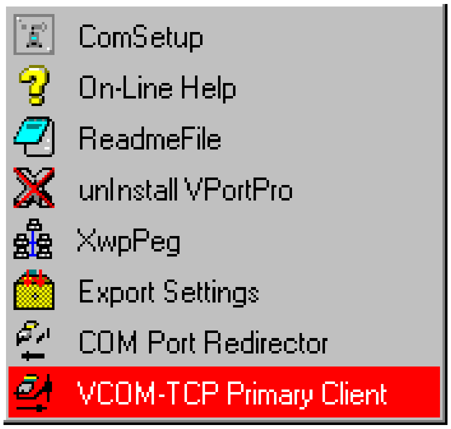
Currently, **VCOM-TCP Primary Client** supports only its own virtual serial ports (i.e. only those it creates by itself). **VCOM-TCP Primary Client** does not support physical serial ports and virtual serial ports created by software tools other than VPortPro.

Currently, **VCOM-TCP Primary Client** cannot redirect a serial port to more than one remote TCP/IP port, and a remote TCP/IP port to more than one serial port.

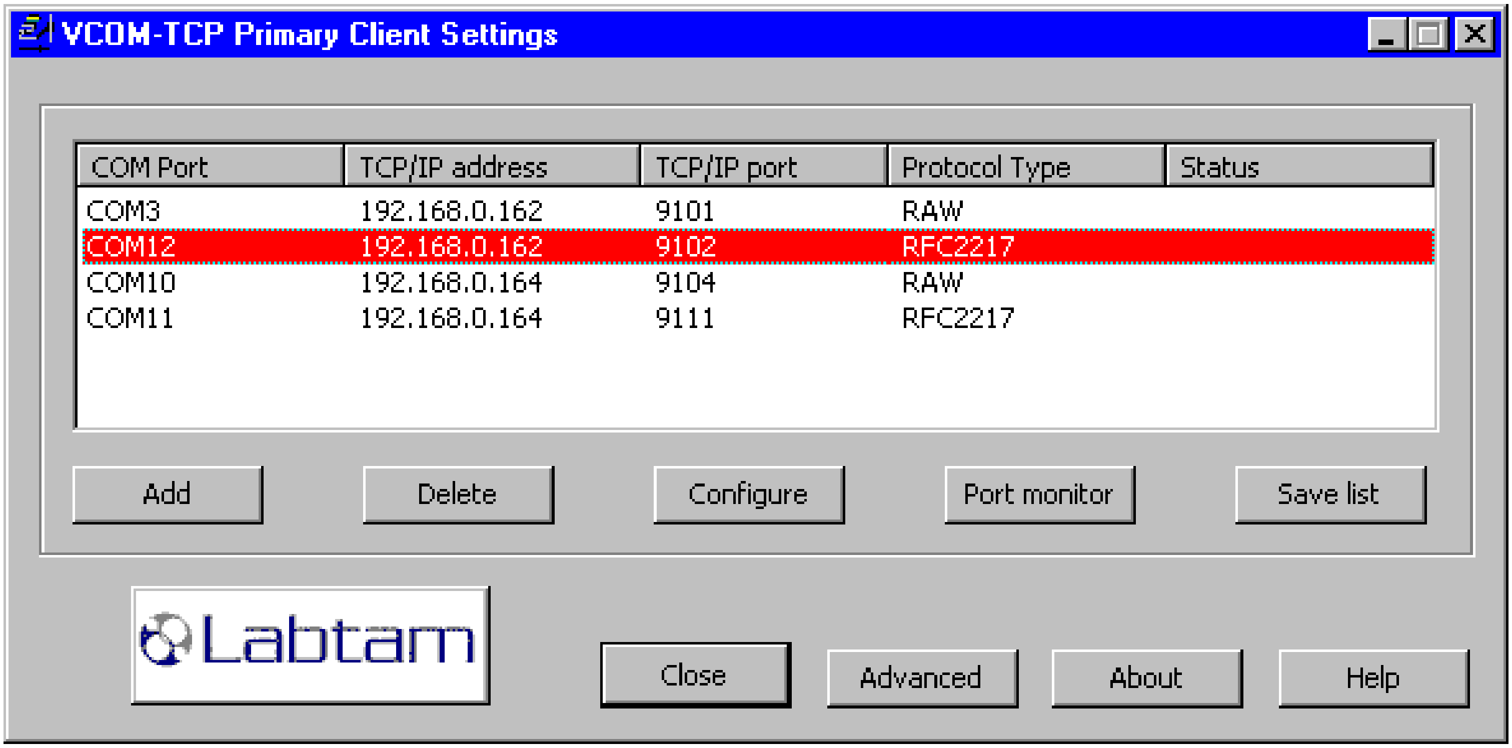
< **previous** | **content** | **next** >

Using VCOM-TCP Primary Client

You can start the program by clicking on the **VCOM-TCP Primary Client** item in the VPortPro Programs' folder (i.e., from the Start/Programs/VPortPro menu):



The **VCOM-TCP Primary Client** dialog box will appear on your display:



COM Port	TCP/IP address	TCP/IP port	Protocol Type	Status
COM3	192.168.0.162	9101	RAW	
COM12	192.168.0.162	9102	RFC2217	
COM10	192.168.0.164	9104	RAW	
COM11	192.168.0.164	9111	RFC2217	

The dialog displays the available connection channels list. The **Status** column shows activity of each connection channel along with the **COM Port** number, the **TCP/IP address** of a serial server, the **TCP/IP port** number of a serial server, and the **Protocol Type** of a connection (**RAW** TCP connection or Telnet (**RFC 2217**) protocol).

Pressing the **Close** button closes the dialog and exits the program with saving settings you have made.

You can press the **Save list** button to write out the connection channel list to an ASCII file (e.g., to document the list or to port it).

The **Advanced** button lets you specify your preferences for all connection channels. (Refer to section **The Advanced Box** below.)

The **Add** button lets you specify initial parameter values for a COM port that you want to add to the connection channel list along with the other settings for the cannel. (Refer to section **Adding a Connection Channel** below.)

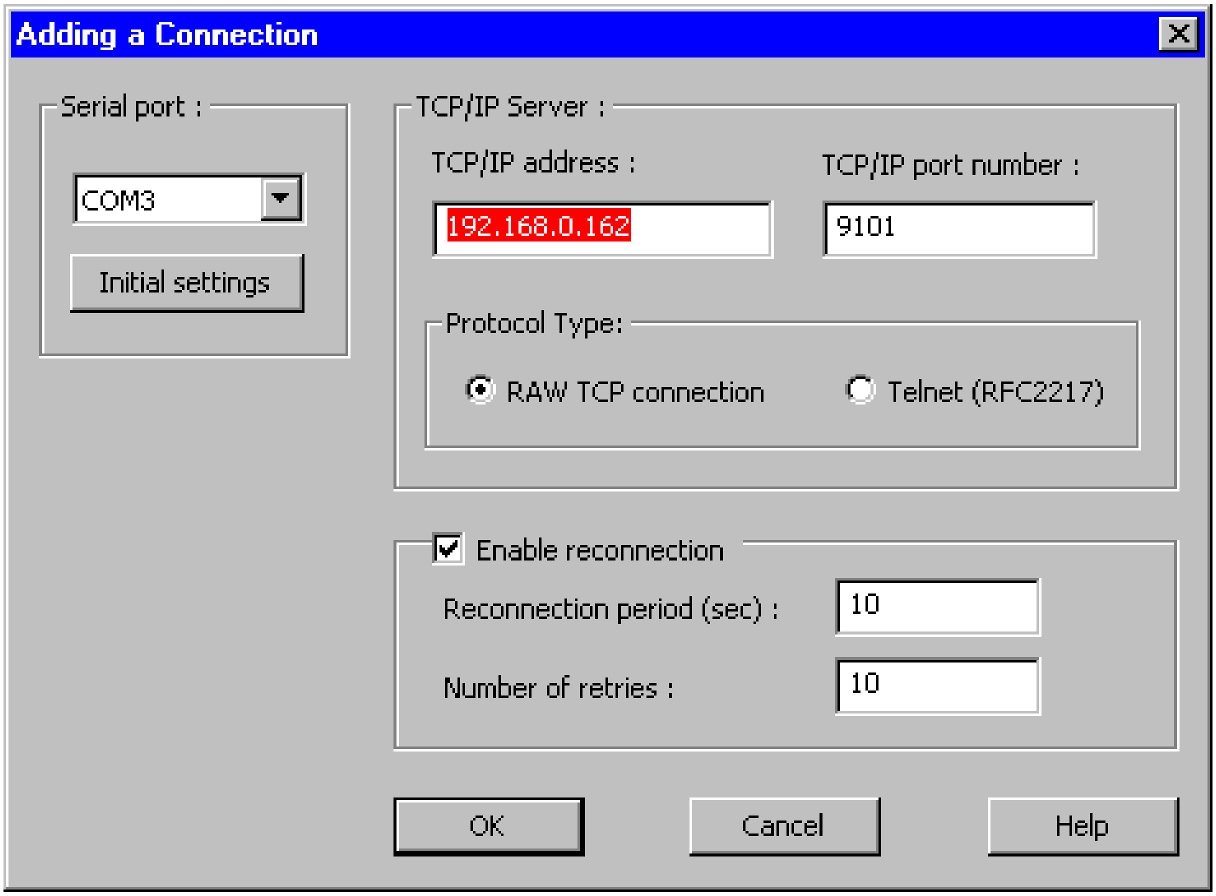
If you select a virtual **COM Port**, then you can **Delete** it from the list (and from your system), **Configure** it (i.e. change settings for it), or display **Port monitor** to watch its activity. (Refer to sections **Configuring a Connection Channel** and **Monitoring COM Ports** below.)

As soon as a new connection channel updates the connection channel list (after **Add** or **Configure**), that channel becomes active immediately. This means that **VCOM-TCP Primary Client** creates virtual serial ports that are available to PC applications at all times (until you **Delete** them from the list), even if no user is logged in. You can use the **VCOM-TCP Primary Client** to verify immediately that it can communicate with the serial server over the network.

After you have configured **VCOM-TCP Primary Client**, you should configure your target applications to use proper virtual serial ports.

Adding a Connection Channel

When you click the **Add** button in the **VCOM-TCP Primary Client** window the **Adding a Connection** dialog brings up:



Serial port : COM3

TCP/IP address : 192.168.0.162

TCP/IP port number : 9101

Protocol Type: ● RAW TCP connection ○ Telnet (RFC2217)

☑ Enable reconnection

Reconnection period (sec) : 10

Number of retries : 10

This dialog allows you to specify a connection channel that you want to add to the connection channel list. A connection channel is made up of a serial port number, its initial settings, a TCP/IP address and TCP/IP port number of a serial server, and a transport data stream protocol type.

To add a new connection channel:

1. In the **Serial port** list box, select a free serial port
2. Make **Initial settings** for the serial port parameters
3. Specify a **TCP/IP address** of a server
4. Specify a **TCP/IP port number** of the server
5. In the **Protocol Type** box, choose a transport data stream protocol type

6. Press **OK**

Your settings become effective just when you click the **OK** button.

In the **Serial Port** list box, you can select from the available virtual ports list a serial port for the connection channel you want to add. Then you can use the **Initial settings** button to specify the configuration parameters for the current COM-port to their default or desired values. Serial ports that already exist in the MS Windows (such as COM1 for a local serial port) are automatically excluded from the available virtual ports list to prevent conflicts.

In the **TCP/IP Server** box, you must fill in the **TCP/IP address** and **TCP/IP port number** of a serial server that you want to connect to for the connection channel.

You can specify any port number between 1 and 65535 which does not cause conflicts between services and **VCOM-TCP Primary Client**. To prevent conflicts between services and **VCOM-TCP Primary Client**, it is recommended to use free port numbers between 1024 and 65535, i.e. unprivileged ports and not used by services. (See also the **services** system file.) A PC application employs ports on a serial server when it uses the COM ports that are created and managed by the **VCOM-TCP Primary Client**.

The Protocol Type

In the **Protocol Type** box, you must choose a transport data stream protocol type that the **VCOM-TCP Primary Client** will use in that connection with the serial server. For connection protocol, the setting must match the TCP/IP protocol that the serial server supports.

Most MS Windows applications perform only the common read/write operations that all serial servers support. Some applications, however, also require some of the features provided by the COM Port Control protocol specified by IETF RFC 2217. For these applications, the serial server must support the COM Port Control protocol and provide at least the serial port control and status functions that the application requires.

There are two options for the transport data stream between the **VCOM-TCP Primary Client** and the serial server:

Raw TCP connection

Raw TCP connection is used to communicate with a serial server without any additional protocol. Although this is possible with most servers, it is not recommended because it precludes the use of helpful Telnet (RFC 2217) protocol features.

Telnet (RFC 2217)

Telnet is the correct setting for most serial servers. If this protocol is selected, the **VCOM-TCP Primary Client** will automatically request the use of the Telnet "binary mode" to allow proper operation of applications. If the serial server supports the COM Port Control protocol (RFC 2217), it will automatically be used.

General TCP/IP settings are described in section **Using ComSetup** of chapter **Configuring VPortPro**. For more information, refer to **Appendix A. TCP/IP Connection Overview**.

Reconnection Settings

With the **Enable reconnection** check box enabled, the **VCOM-TCP Primary Client** will initiate periodical attempts to reconnect the TCP/IP client connection that gets closed. If the connection fails, the connection attempt will be repeated for a user-defined number of times and interval. If the connection attempt fails beyond the user-defined limit, then a pop-up message will appear informing the user that the TCP/IP port is not available and ask if the user wants to keep retrying (the message can be disabled if desired).

This option is useful when any disruption in connection will cause the **VCOM-TCP Primary Client** to automatically go to stop. If the option is disabled, the **VCOM-TCP Primary Client** does not restart.

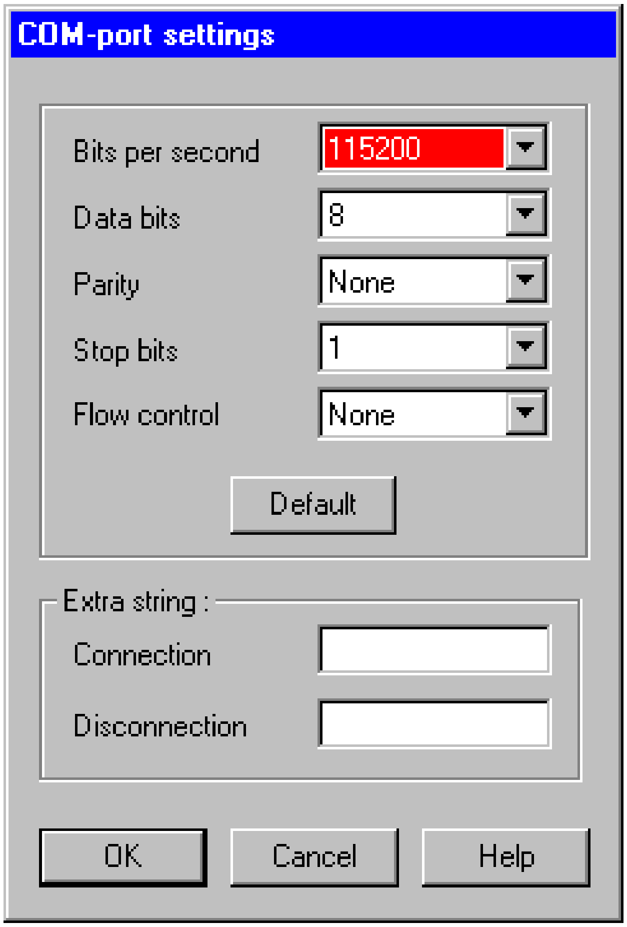
The **Reconnection period** setting specifies the frequency of these attempts (in seconds). The default value is 10.

The **Number of retries** setting specifies the maximal number of these attempts. The default value is 10. The maximal value is 1.000.000. You can enter value 0 for a very large number.

< **previous** | **content** | **next** >

COM-port Settings

When you click the **Initial settings** button in the **Add** dialog the **COM-port settings** dialog box brings up:



COM-port settings

Bits per second: 115200
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

This dialog is used to set up the configuration parameters for the current COM port to their default or desired values (these are initial values because they may be changed when establishing connection in the Telnet RFC-2217 protocol mode). These are **Bits per second**, **Data bits**, **Parity**, **Stop bits**, and **Flow control**. For COM port options, the settings must match the COM port behavior expected by the PC application that will use this COM port.

Default

This button is used to reset the configuration parameters for the current COM-port to their default values.

For more information, refer to **Appendix B. Serial Connection Overview**.

Bits per second

Specifies the transmission rate in bits per second. You can choose an appropriate value from the list box. The default value is 9600.

The speed at which devices transmit data is called the throughput. This is measured in bits per second (bps). Bits per second is the number of bits transmitted every second, used as a measure of the speed at which a device can transfer data. A character is made up of 8 bits. In asynchronous communication, each character may be preceded by a start bit and may terminate with a stop bit. So for each character, 10 bits are transmitted. For example, if a device communicates at 2,400 bits per second, then 240 characters are sent every second.

For serial communications, speed in bits per second can be divided by 10 to approximate the characters transmitted per second.

Data bits

Specifies the number of data bits in a character. Most systems now use 8 bits to represent a single data character (extended ASCII). In rare instances, some older systems still use 7 bits. Valid values for data bits are in the range 5 through 8. The default value is 8. Not all computers support the values 5 and 6. You can choose an appropriate number from the list box.

Parity

Specifies how the system uses the parity bit to check for transmission errors.

The parity value can be set to one of the following:

None

Send no parity bit.

Even

Set the parity bit to 0 or 1 to make the number of 1 bits even.

Odd

Set the parity bit to 0 or 1 to make the number of 1 bits odd.

Mark (1)

Set the parity bit always to 1.

Space (0)

Set the parity bit always to 0.

The default value is **None**. You can choose an appropriate value from the list box. Not all computers support the values **Mark** and **Space**. Most modem connections now use more reliable and sophisticated methods of error checking, so this is usually set to **None**.

Stop bits

Specifies the number of stop bits that define the end of a character: 1, 1.5, or 2. You can choose an appropriate number from the list box. The default value is 1. If the baud rate is 110, the default value is 2; otherwise, the default value is 1. Not all computers support the value 1.5.

Flow control

From within the list box, you can choose one of the following:

XON / XOFF

Specifies whether the XON or XOFF protocol for data-flow control is on or off.

Software flow control (also called XON/XOFF or CTRL+S/CTRL+Q) uses data characters to indicate that the data flow should start or stop. This enables a device to send a control character to signal another device to stop transmitting while it catches up.

Software flow control is slower and usually less desirable than hardware flow control. Software flow control is used only for transmitting text. It cannot be used for binary file transfer because binary data may contain the special flow control characters.

Hardware

Hardware flow control (RTS/CTS) depends on the device to control the flow of data. This should be used with all high-speed modems or modems that compress data.

RTS specifies whether the Request To Send circuit is set to on, off, handshake, or toggle.

CTS specifies whether output handshaking that uses the Clear To Send circuit is on or off.

None

No flow control options will be used (the default value).

Note

When you want to "connect" a serial data source with a serial data destination (e.g., across the network or as two virtual serial ports), make sure that the serial port settings on both sides of such a "connection" be matched.

The Extra String Box

In the **Extra String** box, you can specify the following edit fields:

Connection

When the TCP/IP connection gets established, the bytes sequence you specified in the field will first be sent to the COM port.

Disconnection

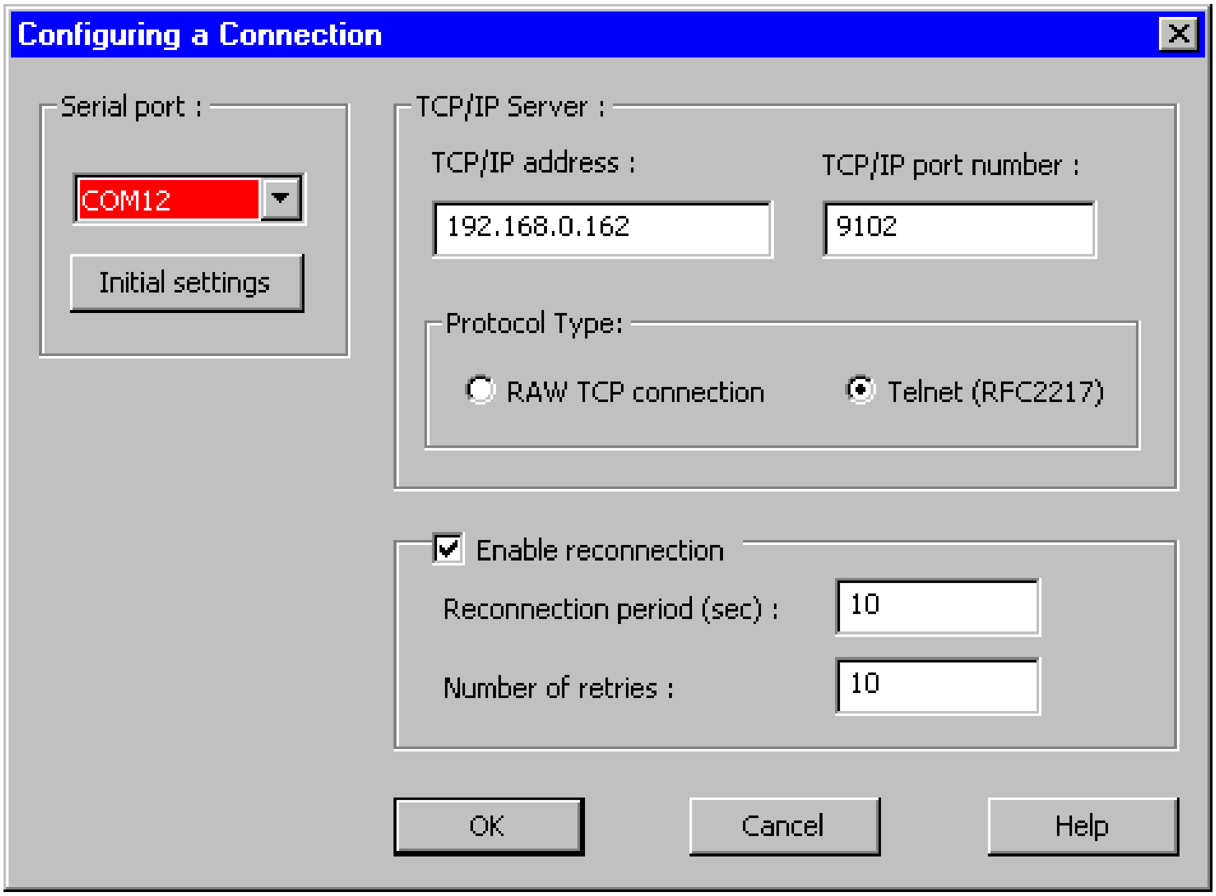
When the TCP/IP connection gets closed, the bytes sequence you specified in the field will finally be sent to the COM port.

These options may be useful if the TCP/IP client is suddenly disconnected from the TCP/IP server while attached via VCOM-TCP Primary Client to one of remote sites. To disconnect the modem from the remote site on the unattended server, this is a way to be able to drop DTR, etc.

< previous | content | next >

Configuring a Connection Channel

When you select a connection channel in the **VCOM-TCP Primary Client** window and click the **Configure** button, the **Configuring a Connection** dialog brings up:



Configuring a Connection

Serial port : COM12

TCP/IP Server :

TCP/IP address : 192.168.0.162

TCP/IP port number : 9102

Protocol Type: ○ RAW TCP connection ● Telnet (RFC2217)

☑ Enable reconnection

Reconnection period (sec) : 10

Number of retries : 10

Each connection channel has its own settings displayed in the dialog to reflect the settings you have made for that COM port.

This dialog lets you change parameter values for a connection channel selected from the connection channel list. These parameters are similar to those displayed in the **Add** dialog box. (Refer to section **Adding a Connection Channel** above.)

Additionally, you can specify whether **VCOM-TCP Primary Client** will restart (according to reconnection parameters you can enter) or not when the connection fails.

To configure an existing connection channel:

1. In the **VCOM-TCP Primary Client** window, select a connection channel
2. Make **Initial settings** for the serial port parameters
3. Specify a **TCP/IP address** of a server
4. Specify a **TCP/IP port number** of the server
5. In the **Protocol Type** box, choose a transport data stream protocol type
6. Press **OK**

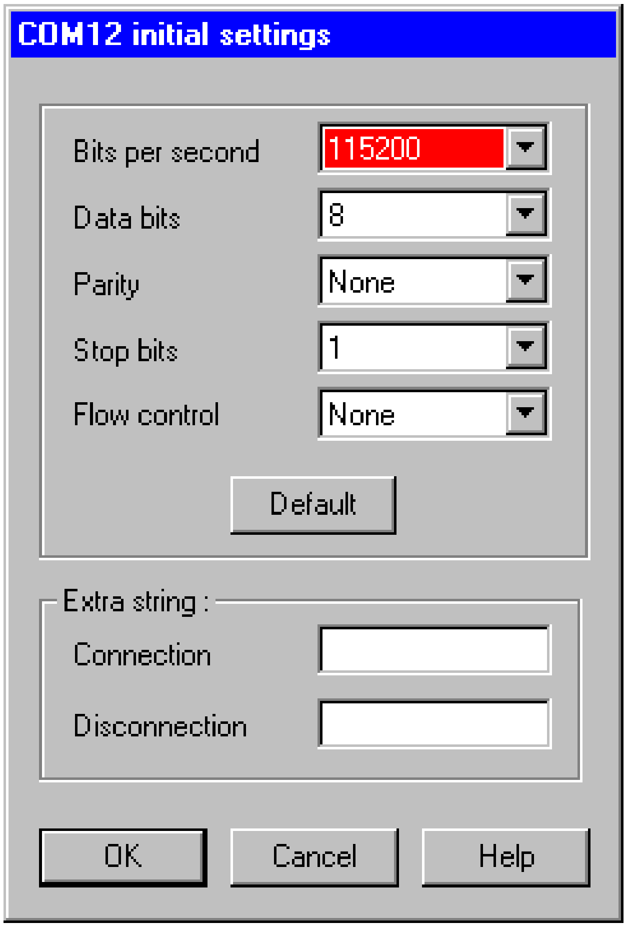
Your changes become effective when you click the **OK** button.

6. VCOM-TCP Primary Client < **previous** | **content** | **next** >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

COM-port Settings

When you click the **Initial settings** button in the **Configure** dialog the **COM-port initial settings** dialog box brings up displaying the COM port number in its header.



COM12 initial settings

Bits per second: 115200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Extra string :

Connection:

Disconnection:

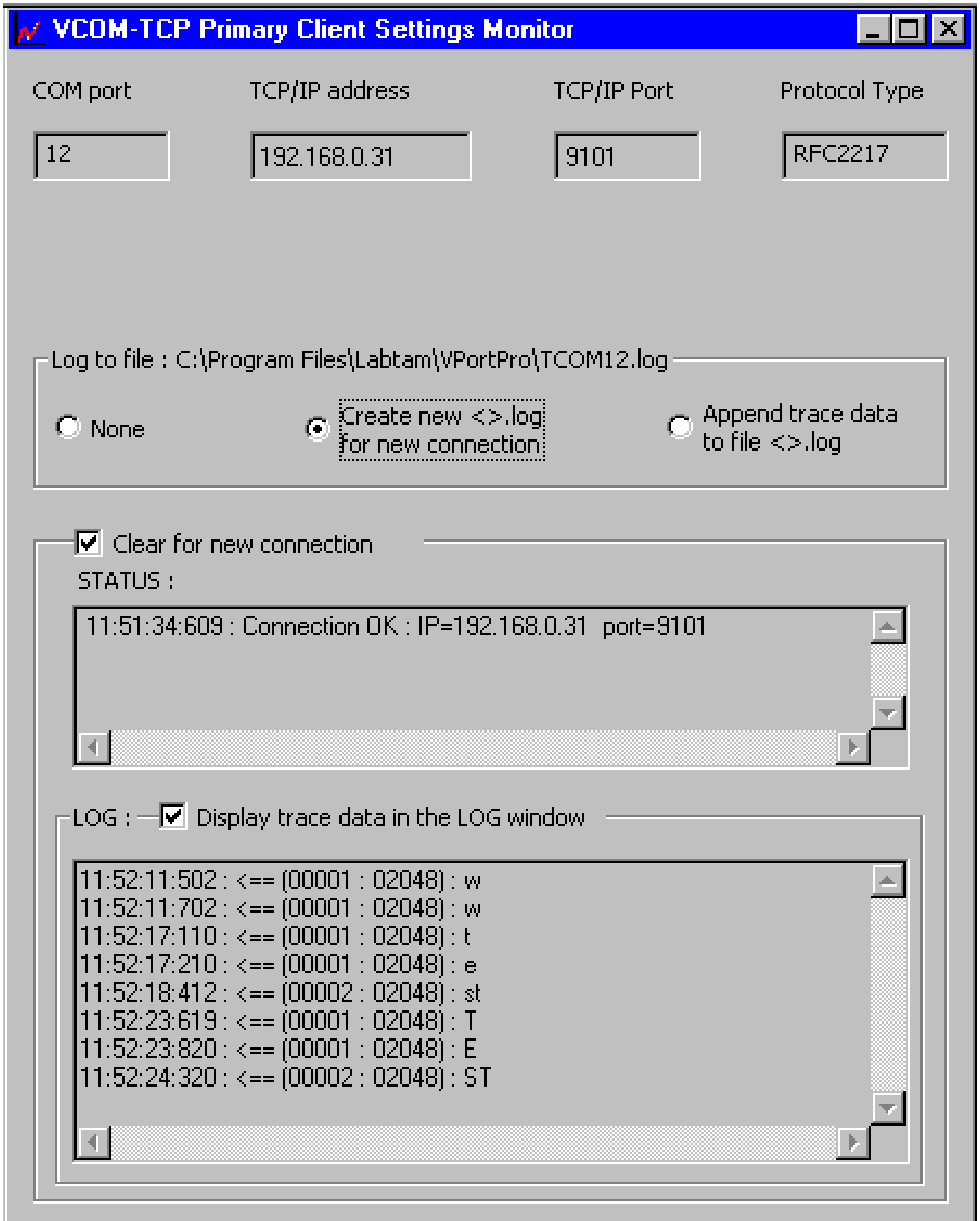
This dialog is used to change the configuration parameters for the selected COM port to their default or desired values (these are initial values because they may be changed when establishing connection in the Telnet RFC-2217 protocol mode). These are **Bits per second**, **Data bits**, **Parity**, **Stop bits**, and **Flow control**. For COM port options, the settings must match the COM port behavior expected by the PC application that will use this COM port.

These parameters are similar to those displayed in the **COM-port Settings** dialog box. (For more information, see subsection **COM-port Settings** in section **Adding a Connection Channel** above.)

< **previous** | **content** | **next** >

Monitoring COM Ports

From the **VCOM-TCP Primary Client** window, you can select a connection channel and press the **Port monitor** button to display the **VCOM-TCP Primary Client Monitor** window. Then you can watch the COM port's activity and control data logging for that channel.



VCOM-TCP Primary Client Settings Monitor

COM port	TCP/IP address	TCP/IP Port	Protocol Type
12	192.168.0.31	9101	RFC2217

Log to file : C:\Program Files\Labtam\VPortPro\TCOM12.log

○ None ● Create new <>.log for new connection ○ Append trace data to file <>.log

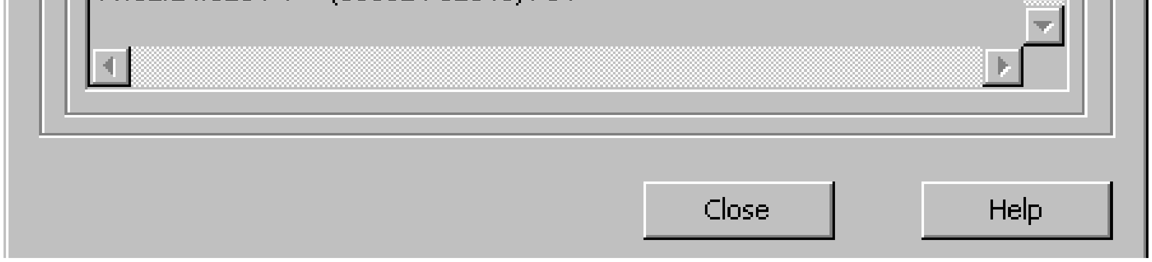
☑ Clear for new connection

STATUS :

```
11:51:34:609 : Connection OK : IP=192.168.0.31  port=9101
```

LOG : ☑ Display trace data in the LOG window

```
11:52:11:502 : <== (00001 : 02048) : w
11:52:11:702 : <== (00001 : 02048) : w
11:52:17:110 : <== (00001 : 02048) : t
11:52:17:210 : <== (00001 : 02048) : e
11:52:18:412 : <== (00002 : 02048) : st
11:52:23:619 : <== (00001 : 02048) : T
11:52:23:820 : <== (00001 : 02048) : E
11:52:24:320 : <== (00002 : 02048) : ST
```



COM port displays a virtual serial port number for the connection channel.

TCP/IP address displays a TCP/IP address of a server for the connection channel.

TCP/IP Port displays a TCP/IP port of the server for the connection channel.

Protocol Type displays a protocol type used for the connection channel.

Close closes the **VCOM-TCP Primary Client Monitor** dialog box and the COM port trace log-file.

The Log to file Box

By using the following radio buttons, you can choose one of trace options you prefer:

None

Specifies that no log-data will be stored in the log-file shown.

Create new <>.log for new connection

Specifies to create a new log-file with the name shown to store new log-information for the connection channel seen under the header of the dialog box. The log-file name reflects the serial port number (as a suffix).

Append trace data to file <>.log

Specifies to append new log-information to the existing log-file (with the name shown) for the connection channel seen under the header of the dialog box. The log-file name reflects the serial port number (as a suffix).

The Clear for New Connection Box

When enabled, **Clear for new connection** cleans up the STATUS and LOG fields when a new connection is established.

STATUS

In this field, information about the status of the connection channel is displayed.

The LOG Box

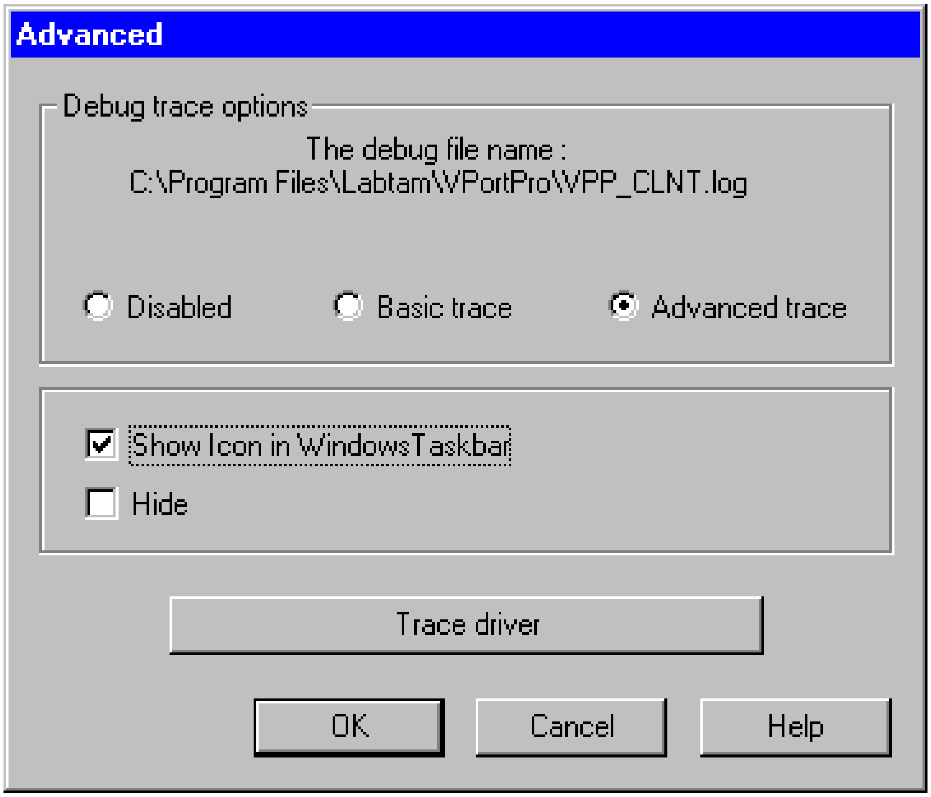
With the **Display trace data in the LOG window** check box enabled, log-information for the connection channel will be displayed in the LOG window below.

< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

The Advanced Box

Clicking the **Advanced** button in the **VCOM-TCP Primary Client** window will come up with the **Advanced** dialog box that lets you specify your preferences for all connection channels in the list:



Advanced

Debug trace options

The debug file name :
C:\Program Files\Labtam\VPortPro\VPP_CLNT.log

○ Disabled ○ Basic trace ● Advanced trace

☑ Show Icon in WindowsTaskbar

□ Hide

By using the following check boxes, you can choose whether the **VCOM-TCP Primary Client** window will be on the desktop or hidden on the system taskbar:

Show Icon in Windows Taskbar

Specifies to place the **VCOM-TCP Primary Client** icon on the system taskbar. Using this icon, you can **Exit** the program, or **Configure** it (i.e. to restore displaying the **VCOM-TCP Primary Client** window).

Hide

With the **Show Icon in Windows Taskbar** check box enabled, specifies to hide the **VCOM-TCP Primary Client** window from the desktop.

Clicking **Trace driver** button calls the tracing utility described below.

The Debug Trace Options Box

By using the following radio buttons, you can choose one of debug trace options you prefer:

Disabled

Specifies that no debug data will be stored into the debug file shown.

Basic trace

Specifies to store brief debug information into the debug file shown.

Advanced trace

Specifies to store full debug information into the debug file shown.

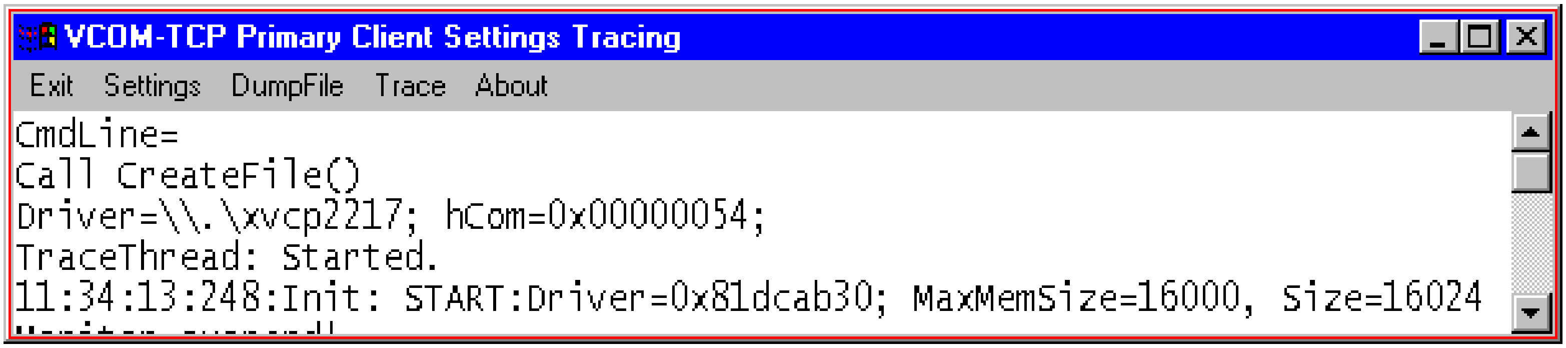
Trace Settings

The tracing utility is designed for setting a number of trace flags (by toggling related check boxes to filter out data of your interest) and passing them to the virtual serial port driver. According to the Trace Flags set, utilities of the package will filter out and save related information into corresponding trace files you can analyze later. The tracing could be used to document the virtual serial port traffic between your software applications that communicate with serial servers via connection channels.

Normally, users should not arbitrarily turn on/off Trace Flags. The flags can noticeably slow down the traffic and can generate a great amount of data. The tracing utility is primarily meant for testing and analyzing how your virtual serial port software applications interface to access servers while using serial port communication features.

Driver Tracing Settings

You can start the tracing program by clicking the **Trace driver** button in the **Advanced** dialog box. The **VCOM-TCP Primary Client Settings Tracing** window will appear on your display:



```
CmdLine=
Call CreateFile()
Driver=\\.\xvcp2217; hCom=0x00000054;
TraceThread: Started.
11:34:13:248:Init: START:Driver=0x81dcab30; MaxMemSize=16000, Size=16024
```

In this window, the program displays event tracing information according to trace flags you specified for the virtual serial port driver in the **Trace Settings** dialog box described below. The information displayed will also be saved in a log file (if specified in the **Trace Settings** dialog box).

Pressing **Exit** on the menu bar closes the window and exits the program.

Pressing **Settings** on the menu bar brings up the **Trace Settings** dialog box. (Refer to section **Making Trace Settings** below.)

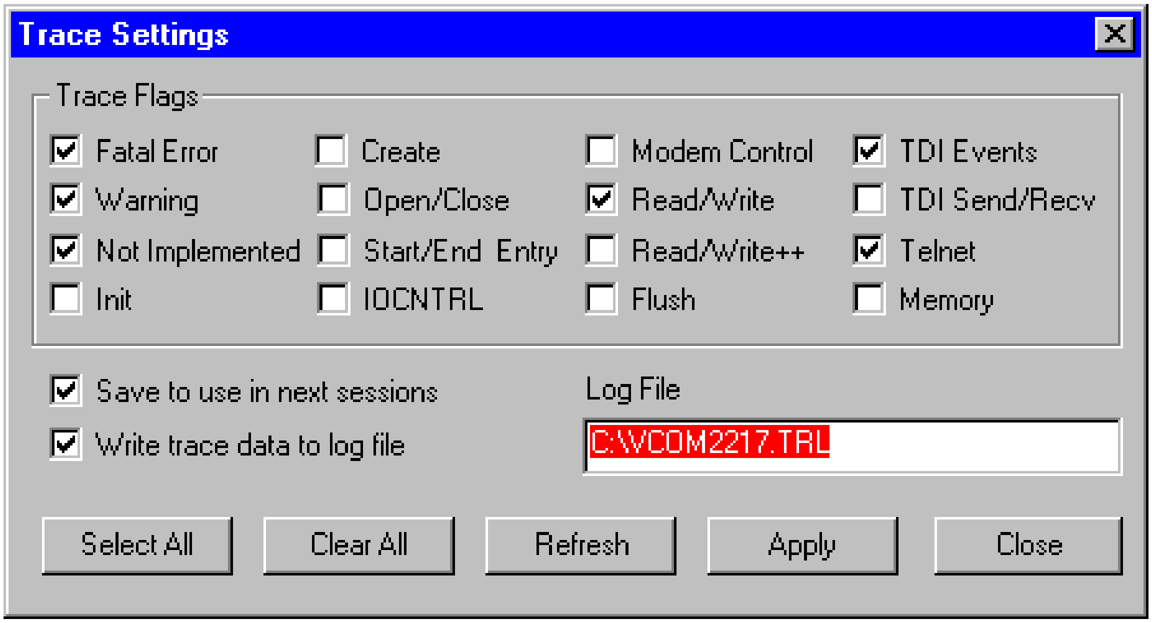
Pressing **DumpFile** on the menu bar brings up the **Dump File** dialog box. (Refer to section **The**

Dump File Box below.)

Pressing **Trace** on the menu bar brings up a submenu where you can choose to start or stop tracing.

Making Trace Settings

Pressing **Settings** on the menu bar of the **VCOM-TCP Primary Client Settings Tracing** window brings up the **Trace Settings** dialog box:



☑ Fatal Error
☑ Warning
☑ Not Implemented
□ Init
□ Create
□ Open/Close
□ Start/End Entry
□ IOCNTRL
□ Modem Control
☑ Read/Write
□ Read/Write++
□ Flush
☑ TDI Events
□ TDI Send/Recv
☑ Telnet
□ Memory

☑ Save to use in next sessions
☑ Write trace data to log file

Log File: C:\VCOM2217.TRL

If you click the **Close** button, the program closes the dialog with no changes.

Pressing the **Apply** button passes the Trace Flags set to the virtual serial port driver. The Trace Flags will take effect immediately.

Select All makes all the Trace Flags enabled. **Clear All** makes all the Trace Flags disabled.

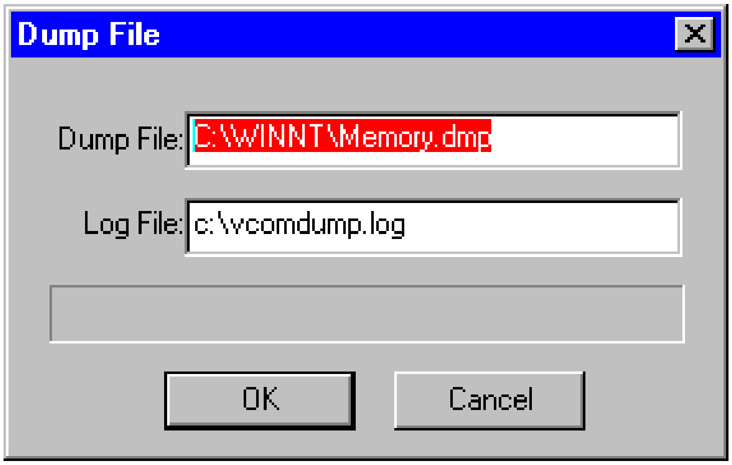
Refresh restores all Trace Flags to their initial states (when loaded the tracing utility or after the latest **Apply**).

With the **Save to use in next sessions** check box enabled, you can store the current settings (by pressing **Apply**) and load them later (when loading the tracing utility).

With the **Write trace data to log file** check box enabled, you can enter a full path file name or leave the default one in the **Log File** field to store flagged log information into that file.

The DumpFile Box

The **DumpFile** on the menu bar of the **VCOM-TCP Primary Client Settings Tracing** window brings up the dialog box:



Dump File: C:\WINNT\Memory.dmp

Log File: c:\vcomdump.log

You can specify a full path name for one of system dump files to be read (when system crash) and a log file name to store the driver tracing information extracted from that dump file that you can analyze later. Normally, you can use the default file names shown.

< previous | content | next >

7. COM Port Redirector

This chapter describes how to start and use the **COM Port Redirector** program supplied with VPortPro.

The **COM Port Redirector** is a software program designed to convert serial (RS232) data to the TCP/IP packets format and TCP/IP packets to serial data. While running on your PC, **COM Port Redirector** can receive serial data from a serial port of your PC and then send them out to a TCP/IP network through a TCP/IP socket connection on your PC. **COM Port Redirector** can also receive serial data from a TCP/IP network through a TCP/IP socket connection on your PC and send them to a serial port on your PC.

COM Port Redirector supports two transport data stream protocols:

- Raw TCP connection (without sending configuration or status information from the PC to the serial device or server)
- Telnet (with the features provided by the COM Port Control protocol specified by IETF RFC-2217)

COM Port Redirector can be configured to function as a TCP/IP server and as a TCP/IP client for both the protocol types.

As a TCP/IP client, **COM Port Redirector** will initiate and establish connection with remote systems (by sending connection requests), and then communicate with connected TCP/IP servers. As a TCP/IP client, **COM Port Redirector** allows to establish connection to a single TCP/IP server for each serial port. Normally any data received through the TCP/IP port that it connects to will be transmitted out the serial port and any data received through the serial port will be sent back to the TCP/IP server. This enables fully bi-directional communications. So, **COM Port Redirector** redirects a local COM port to a TCP/IP port of a remote server and vice versa (using a local TCP/IP socket connection).

As a TCP/IP server, it will wait for remote TCP/IP clients' connection requests to TCP/IP ports (you specified as listening on your PC) to establish connection with remote systems, and then to communicate with connected TCP/IP clients. As a TCP/IP server, **COM Port Redirector** allows to accept only a single client connection to a TCP/IP port. Incoming data received from a TCP/IP client is transmitted out a serial port as it is received. Incoming serial data is sent to a connected TCP/IP client. This enables fully bi-directional communications. So, **COM Port Redirector** maps and redirects COM ports to a TCP/IP ports and vice versa.

Also with **COM Port Redirector**, you can create "virtual serial ports". A virtual serial port (VSP) appears to application software as a "standard" MS Window's communications port (virtually identical to an RS-232 port). Data may be written to VSP by applications, and data may be read from VSP as though it were a standard COM port. You can configure **COM Port Redirector** to map TCP/IP ports onto virtual serial ports as well as to physical COM ports. This allows MS Windows applications to generate or consume data to and from VSP as though the data were being processed by physical hardware.

You can use the virtual COM port driver on multi-processor and Hyper-Threading machines too.

While you are using **COM Port Redirector**, you can:

- Configure COM-TCP/IP connection channels (i.e., make COM and TCP/IP mapping and settings for a connection channel)
- Manage the connection channels list (i.e., Add/Delete connection channels)
- Watch connection channels activity (i.e., status)
- Monitor data logging for a selected connection channel from the list
- Trace the **COM Port Redirector** (driver) session using a variety of trace flags
- Connect/Disconnect connection channels (for the Client mode)

Currently, **COM Port Redirector** supports physical serial ports and virtual serial ports as well. Virtual serial ports can be created by COM Port Redirector itself or by software tools other than COM Port Redirector, for example, by VCOM-TCP Primary Client (COM Port Redirector will treat them as physical serial ports).

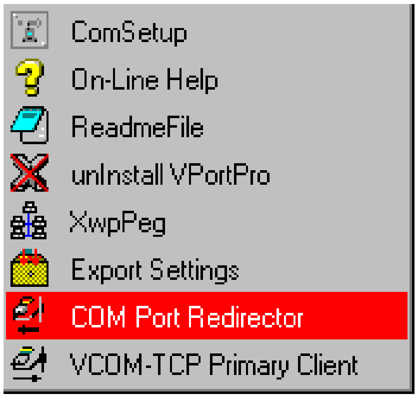
Currently, **COM Port Redirector** cannot redirect a serial port to more than one remote TCP/IP port, and a remote TCP/IP port to more than one serial port.

Also a single virtual serial port cannot simultaneously be used for both Client and Server modes.

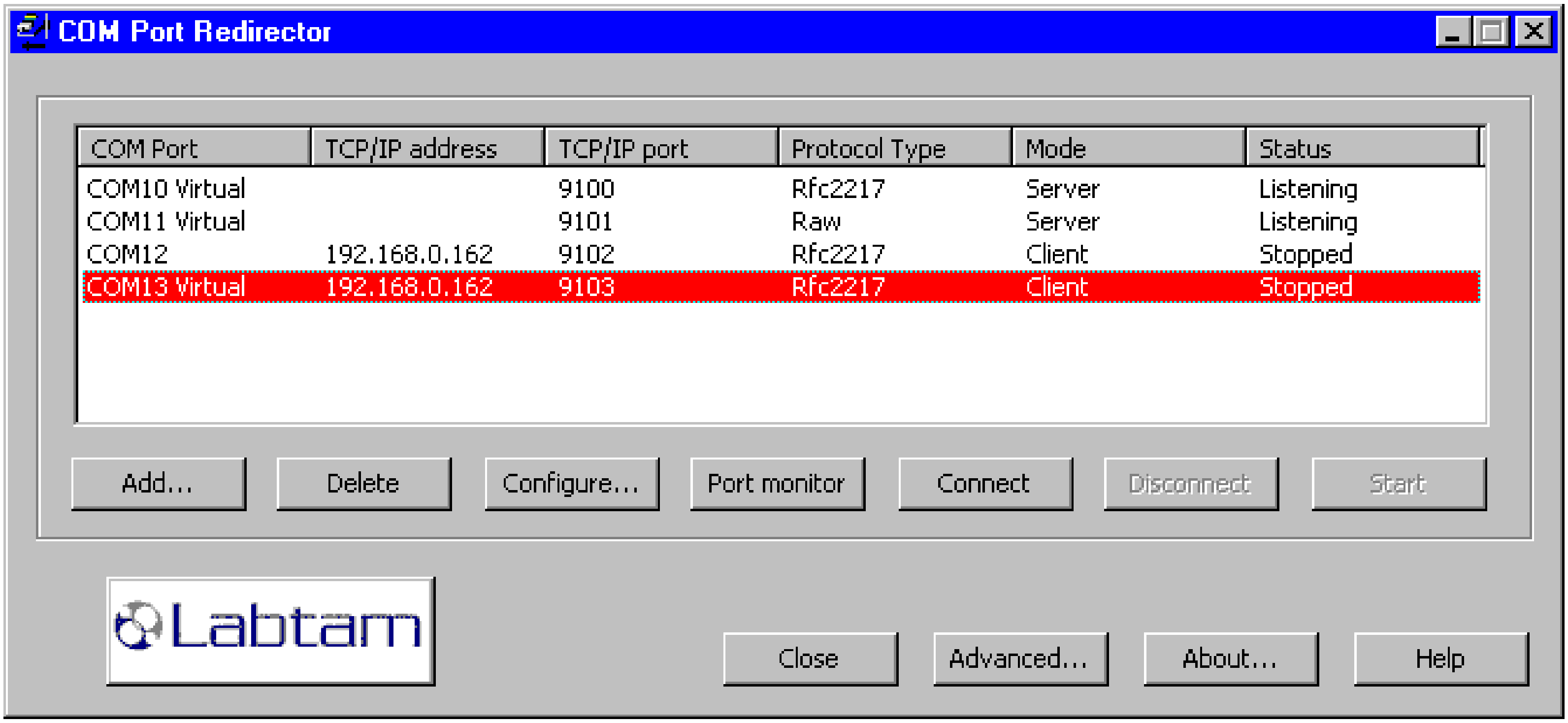
< **previous** | **content** | **next** >

Using COM Port Redirector

You can start the program by clicking on the **COM Port Redirector** item in the VPortPro Programs' folder (i.e., from the Start/Programs/VPortPro menu):



The **COM Port Redirector** window will appear on your display:



COM Port	TCP/IP address	TCP/IP port	Protocol Type	Mode	Status
COM10 Virtual		9100	Rfc2217	Server	Listening
COM11 Virtual		9101	Raw	Server	Listening
COM12	192.168.0.162	9102	Rfc2217	Client	Stopped
COM13 Virtual	192.168.0.162	9103	Rfc2217	Client	Stopped

The window displays the available connection channels list.

For each connection channel, the **Status** column shows activity (listening, stopped, connected), the **COM Port** number, the **TCP/IP address** (of the TCP/IP server for the Client mode), the **TCP/IP port** number, the **Protocol Type** of a connection (**RAW TCP connection** or **Telnet (RFC 2217)** protocol), and the **Mode** of connection (**Client** or **Server**).

The question sign (?) next to COM port numbers marks virtual serial ports that are absent in your system (not created). The word **Virtual** next to COM port numbers marks created virtual serial ports in your system. COM ports with no sign are considered as physical ones (or created by software other than VPortPro).

Pressing the **Close** button closes the window and exits the program with saving settings you have made to the list.

The **Add** button lets you specify parameter values for a COM port that you want to add to the connection channel list along with the other settings for the cannel. (Refer to section **Adding a Connection Channel**.)

If you select a **COM Port** from the list, then you can **Delete** it from the list (and from your system), or **Configure** it (i.e. change settings for it). (Refer to section **Configuring a Connection Channel**.)

You can select one or more unconnected COM ports (seen in the Client mode) and **Connect** them (i.e., establish TCP-connections and make the ports available over the network).

You can **Disconnect** a group of selected connected COM ports from your network (for any mode). In the Client mode, the **Reconnection period** setting specifies the frequency of connection attempts (for your client to be able to connect to the selected server). (See section **Reconnection Settings**.)

The **Connect / Disconnect** actions may be treated as "... CURRENT connection to/from that port".

The **Start / Stop** button lets you start or stop the server's activity on a selected listening port (for remote clients to be able to connect to your server or not). In the Client mode, this button lets you start or stop the client's activity on a selected port (for your client to be able to connect to the selected server or not).

The **Start / Stop** actions must be considered as "... ANY connection to/from that port".

You can click the **Port monitor** button to display the **COM Port Redirector Monitor** window to watch activity of a selected and active COM port. (Refer to section **Monitoring COM Ports**.)

Clicking the **Advanced** button will come up with the **Advanced** dialog box that lets you specify your preferences for all connection channels in the list. The **Advanced** dialog box also lets you run tracing of the virtual serial port driver. (Refer to section **The Advanced Box**.)

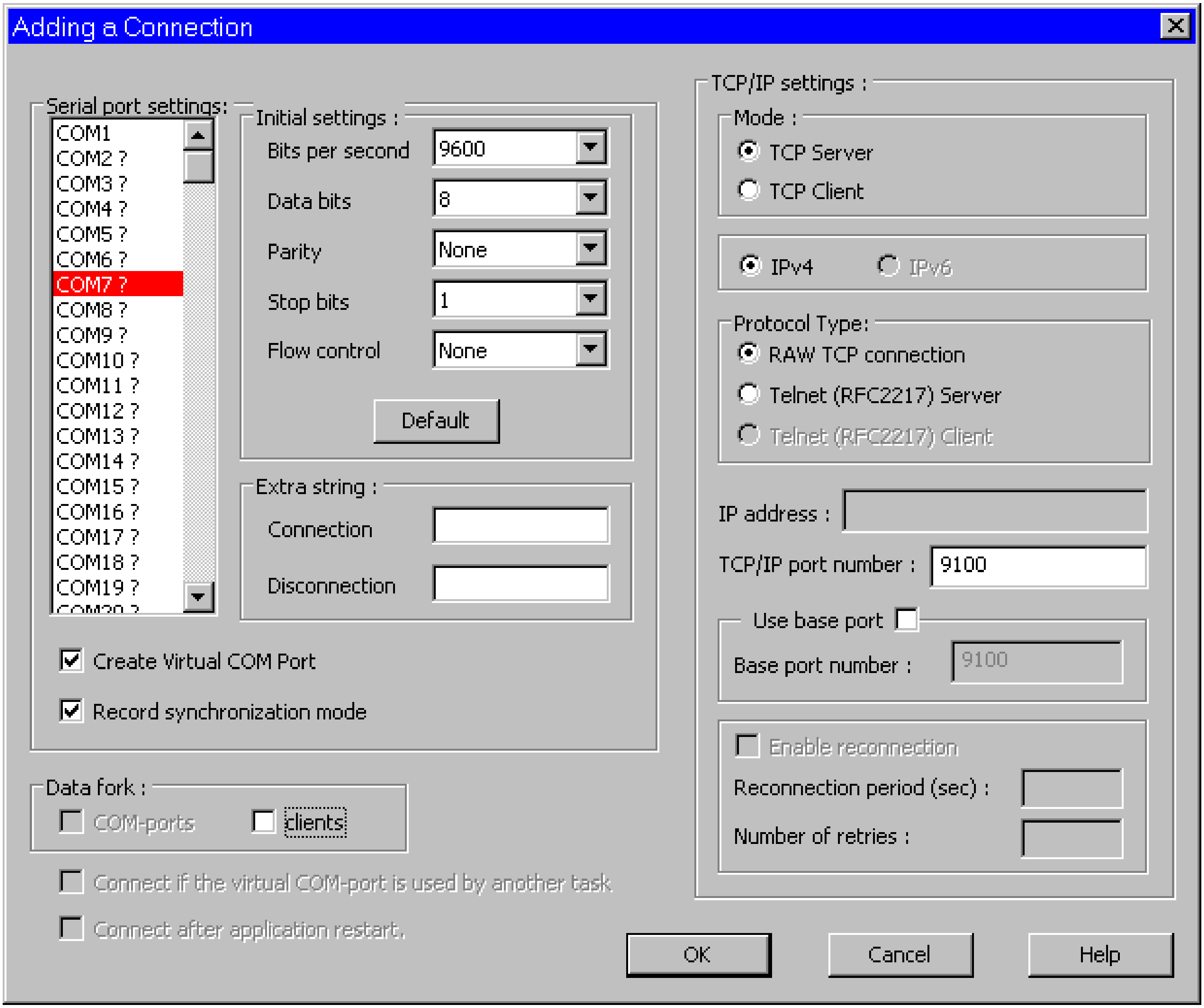
As soon as a new connection channel updates the connection channel list (after **Add** or **Configure**), that channel becomes active immediately. This means that **COM Port Redirector** creates virtual serial ports that are available to PC applications at all times (until you **Delete** them from the list), even if no user is logged in. You can use the **COM Port Redirector** to verify immediately that it can communicate with the serial server over the network.

After you have configured **COM Port Redirector**, you should configure your target applications to use proper virtual serial ports.

< previous | content | next >

Adding a Connection Channel

When you click the **Add** button in the **COM Port Redirector** window the **Adding a Connection** dialog brings up:



Adding a Connection

Serial port settings:
COM1
COM2 ?
COM3 ?
COM4 ?
COM5 ?
COM6 ?
COM7 ?
COM8 ?
COM9 ?
COM10 ?
COM11 ?
COM12 ?
COM13 ?
COM14 ?
COM15 ?
COM16 ?
COM17 ?
COM18 ?
COM19 ?

Initial settings :
Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None
Default

Extra string :
Connection:
Disconnection:

☑ Create Virtual COM Port
☑ Record synchronization mode

Data fork :
□ COM-ports □ clients

□ Connect if the virtual COM-port is used by another task
□ Connect after application restart.

TCP/IP settings :
Mode :
● TCP Server
○ TCP Client

● IPv4 ○ IPv6

Protocol Type:
● RAW TCP connection
○ Telnet (RFC2217) Server
○ Telnet (RFC2217) Client

IP address :
TCP/IP port number : 9100

Use base port □
Base port number : 9100

□ Enable reconnection
Reconnection period (sec) :
Number of retries :

This dialog allows you to specify a connection channel that you want to add to the connection channel list.

A connection channel is made up of a serial port number with its initial settings (grouped in the **Serial port settings** box), and a TCP/IP port number with its initial settings (grouped in the **TCP/IP settings** box).

The question sign (?) next to COM port numbers in the list of available serial ports marks virtual serial ports that are absent in your system (not created and free).

To add a new connection channel:

In the **Serial port settings** group box,

1. Select a free serial port from the list of available serial ports
2. Make **Initial settings** for the serial port parameters (or click the **Default** button)
3. Specify a **Connection Extra string** and/or **Disconnection** one (or leave the fields empty)
4. Enable or disable the **Record synchronization mode** check box
5. Enable or disable the **Create Virtual COM Port** check box

In the **TCP/IP settings** group box,

6. Specify a connection mode (**TCP Client** or **TCP Server**)
7. For the **TCP Server** mode, choose a protocol version (**IPv4** or **IPv6**)
8. Choose a protocol type (**RAW TCP Connection** or **Telnet (RFC2217)**)
9. Specify an **IP address** (for the serial server when in the **TCP Client** mode)
10. Specify a **TCP/IP port number** (for the server to listen to)
11. Specify reconnection parameters (or disable the **Enable reconnection** check box)
12. Press **OK** to put your settings into action

Please refer to section **TCP/IP Settings** below on how to make TCP/IP settings.

Please refer to section **COM Port Settings** below on how to make COM port settings.

Note that a single virtual serial port cannot simultaneously be used for both Client and Server modes.

Connect after application restart

This check box is available for the **TCP Client** mode.
With this option on, COM Port Redirector will automatically try to connect that COM port to its remote server after each restart of COM Port Redirector.
If you disable this check box, then you will have to **Connect** that COM port with its remote server manually (using the button in the **COM Port Redirector** window).

Connect if the virtual COM-port is used by another task

This check box is available for the **TCP Client** mode.
Select this option to set the "auto" mode that provides "VCOM-TCP Primary Client"-like behavior for

COM Port Redirector.
In this "auto" mode, opening/closing a virtual serial port provides the corresponding **Connect/ Disconnect** and **Start/Stop** logic.

The Data fork Box

You can specify how COM Port Redirector as a TCP/IP server will work for the following modes:
- "single client - multiple serial ports",
- "multiple clients - single serial port",
- "multiple clients - multiple serial ports".

- **Clients**
  This option is only available for the server mode of COM Port Redirector and if you have only one serial port seen in the **Port settings list**.

  If the **Clients** check box is off, then COM Port Redirector allows only one network client (one that first connected to COM Port Redirector) to interact with the redirected serial port. Other clients (connected to COM Port Redirector) will wait until freeing the port.

  If the **Clients** check box is selected, then COM Port Redirector will in turn redirect the incoming data flow (i.e., received from each network client connected to COM Port Redirector) to the serial port (intermixing as information goes). Also, when the serial port sends data to the network, COM Port Redirector will in turn duplicate the outgoing data flow to each network client connected to COM Port Redirector.

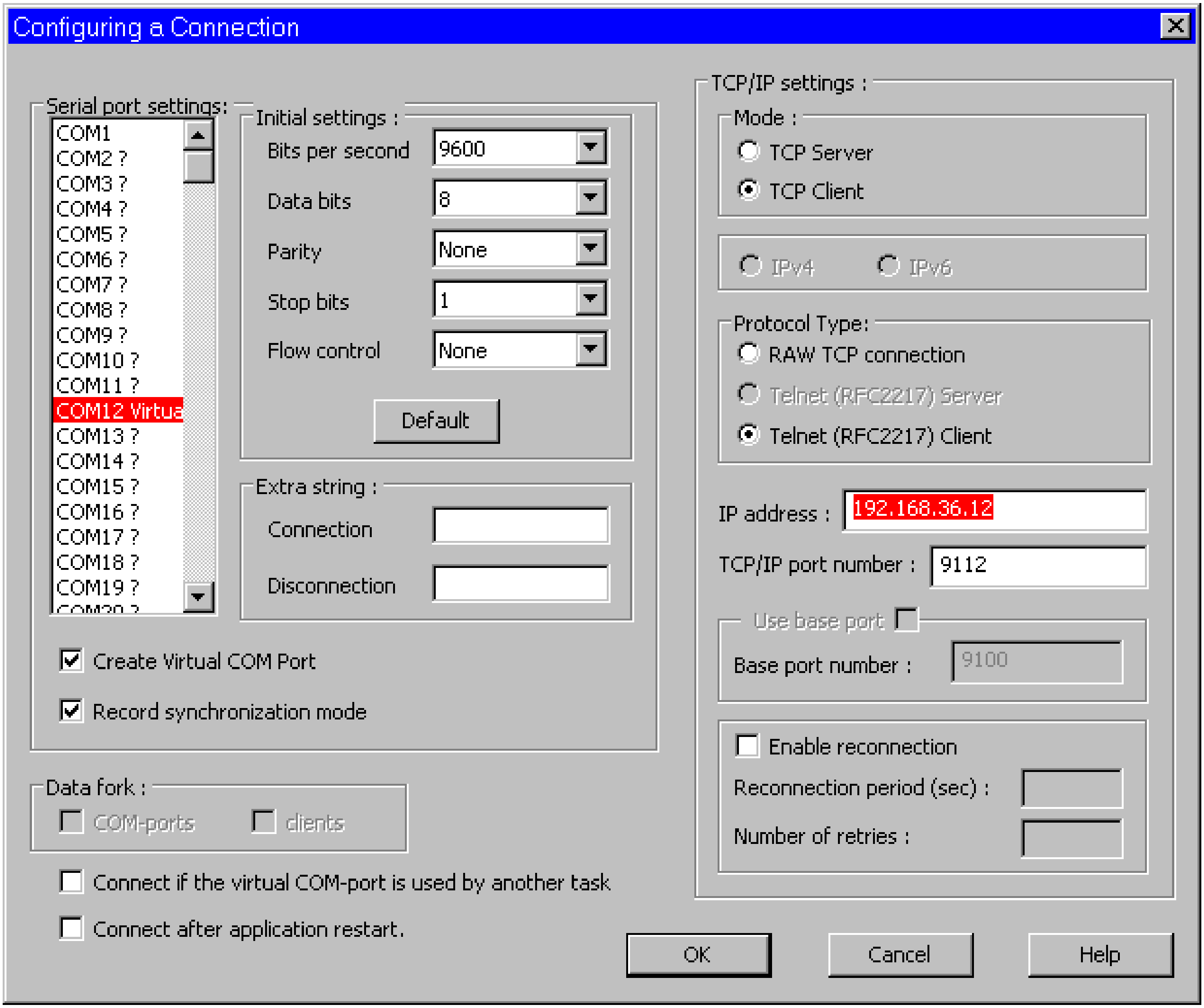
- **COM-ports**
  This option is only available for the server mode of COM Port Redirector and if you have more than one serial port seen in the **Port settings list**.

  If the **COM-ports** check box is off, then COM Port Redirector allows only one network client (one that first connected to COM Port Redirector) to interact with all the redirected serial ports from the list. Other clients (connected to COM Port Redirector) will wait until freeing a port. COM Port Redirector will duplicate the incoming data flow (received from that network client) to all the serial ports in turn. Also, when the serial ports send data to the network, COM Port Redirector will in turn redirect the outgoing data flow to that network client (intermixing as information goes).

  If the **COM-ports** check box is selected, then COM Port Redirector will give all the serial ports from the list to all clients connected to COM Port Redirector (one free port to only one client).

Configuring a Connection Channel

When you select a connection channel in the **COM Port Redirector** window and click the **Configure** button, the **Configuring a Connection** dialog brings up:



Configuring a Connection

Serial port settings:
COM1
COM2 ?
COM3 ?
COM4 ?
COM5 ?
COM6 ?
COM7 ?
COM8 ?
COM9 ?
COM10 ?
COM11 ?
COM12 Virtua
COM13 ?
COM14 ?
COM15 ?
COM16 ?
COM17 ?
COM18 ?
COM19 ?

Initial settings :
Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None
Default

Extra string :
Connection:
Disconnection:

☑ Create Virtual COM Port
☑ Record synchronization mode

Data fork :
□ COM-ports □ clients

□ Connect if the virtual COM-port is used by another task
□ Connect after application restart.

TCP/IP settings :
Mode :
○ TCP Server
● TCP Client

○ IPv4 ○ IPv6

Protocol Type:
○ RAW TCP connection
○ Telnet (RFC2217) Server
● Telnet (RFC2217) Client

IP address : 192.168.36.12
TCP/IP port number : 9112

Use base port □
Base port number : 9100

□ Enable reconnection
Reconnection period (sec) :
Number of retries :

OK Cancel Help

Each connection channel has its own settings displayed in the dialog to reflect the values you have previously made for that channel.

This dialog lets you change parameter values for a connection channel selected from the connection channel list. These parameters are similar to those displayed in the **Adding a Connection** dialog box. (Refer to section **Adding a Connection Channel**.)

Note that a current COM port for the channel is displayed in the serial port list. If you change it, then it will be removed from your system when exiting (with freeing its number if created) and a new COM port with your number will be created.

To configure an existing connection channel:

In the **Serial port settings** group box,

1. Change a COM port number for a connection channel
2. Make **Initial settings** for the serial port parameters (or click the **Default** button)
3. Specify a **Connection Extra string** and/or **Disconnection** one (or leave the fields empty)
4. Enable or disable the **Record synchronization mode** check box
5. Enable or disable the **Create Virtual COM Port** check box

In the **TCP/IP settings** group box,

6. Specify a connection mode (**TCP Client** or **TCP Server**)
7. For the **TCP Server** mode, choose a protocol version (**IPv4** or **IPv6**)
8. Choose a protocol type (**RAW TCP Connection** or **Telnet (RFC2217)**)
9. Specify an **IP address** (for the serial server when in the **TCP Client** mode)
10. Specify a **TCP/IP port number** (for the server to listen to)
11. Specify reconnection parameters (or disable the **Enable reconnection** check box)
12. Press **OK** to put your settings into action

Please refer to section **TCP/IP Settings** below on how to make TCP/IP settings.

Please refer to section **COM Port Settings** below on how to make COM port settings.

Note that a single virtual serial port cannot simultaneously be used for both Client and Server modes.

Connect after application restart

This check box is available for the **TCP Client** mode.
With this option on, COM Port Redirector will automatically try to connect that COM port to its remote server after each restart of COM Port Redirector.
If you disable this check box, then you will have to **Connect** that COM port with its remote server manually (using the button in the **COM Port Redirector** window).

Connect if the virtual COM-port is used by another task

This check box is available for the **TCP Client** mode.

Select this option to set the "auto" mode that provides "VCOM-TCP Primary Client"-like behavior for COM Port Redirector.
In this "auto" mode, opening/closing a virtual serial port provides the corresponding **Connect**/ **Disconnect** and **Start**/**Stop** logic.

The Data fork Box

You can specify how COM Port Redirector as a TCP/IP server will work for the following modes:
- "single client - multiple serial ports",
- "multiple clients - single serial port",
- "multiple clients - multiple serial ports".

- **Clients**
  This option is only available for the server mode of COM Port Redirector and if you have only one serial port seen in the **Port settings list**.

  If the **Clients** check box is off, then COM Port Redirector allows only one network client (one that first connected to COM Port Redirector) to interact with the redirected serial port. Other clients (connected to COM Port Redirector) will wait until freeing the port.

  If the **Clients** check box is selected, then COM Port Redirector will in turn redirect the incoming data flow (i.e., received from each network client connected to COM Port Redirector) to the serial port (intermixing as information goes). Also, when the serial port sends data to the network, COM Port Redirector will in turn duplicate the outgoing data flow to each network client connected to COM Port Redirector.

- **COM-ports**
  This option is only available for the server mode of COM Port Redirector and if you have more than one serial port seen in the **Port settings list**.

  If the **COM-ports** check box is off, then COM Port Redirector allows only one network client (one that first connected to COM Port Redirector) to interact with all the redirected serial ports from the list. Other clients (connected to COM Port Redirector) will wait until freeing a port. COM Port Redirector will duplicate the incoming data flow (received from that network client) to all the serial ports in turn. Also, when the serial ports send data to the network, COM Port Redirector will in turn redirect the outgoing data flow to that network client (intermixing as information goes).

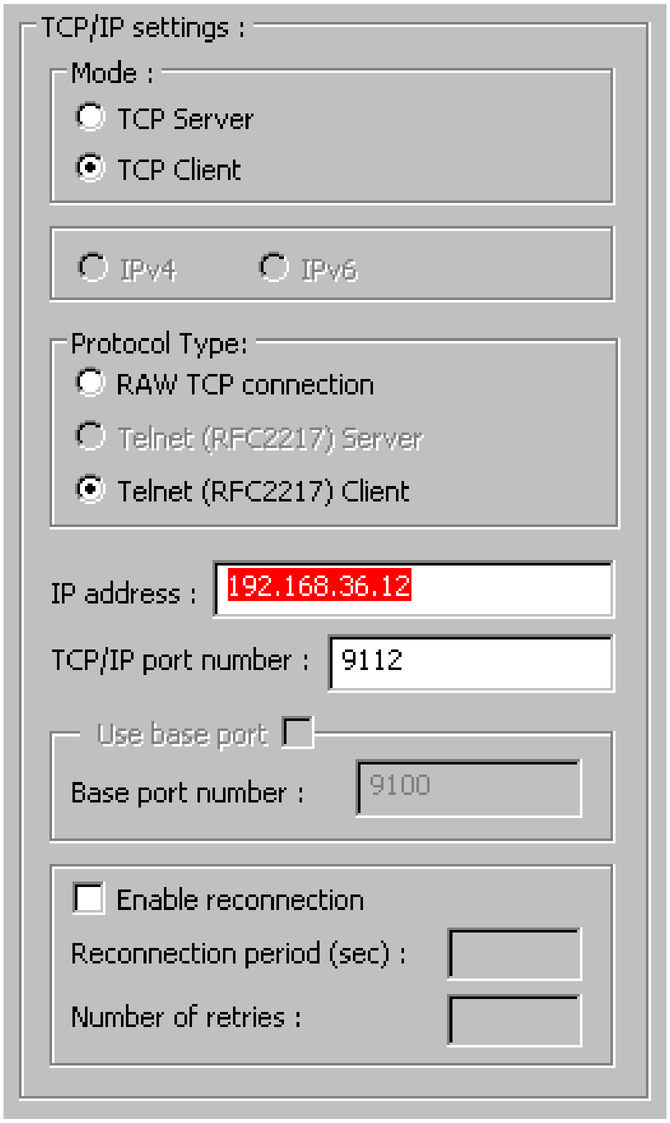
  If the **COM-ports** check box is selected, then COM Port Redirector will give all the serial ports from the list to all clients connected to COM Port Redirector (one free port to only one client).

< previous | content | next >

TCP/IP Settings

In the **TCP/IP settings** group box of the **Adding a Connection** or **Configure a Connection** window, you must specify:

1. A connection mode (**TCP Client** or **TCP Server**)
2. For the **TCP Server** mode, a protocol version (**IPv4** or **IPv6**)
3. A protocol type (**RAW TCP Connection** or **Telnet (RFC2217)**)
4. An **IP address** (for the serial server when in the **TCP Client** mode)
5. A **TCP/IP port number** (for the server to listen to)
6. Reconnection parameters (or disable the **Enable reconnection** check box)



TCP/IP settings :

Mode :

○ TCP Server

● TCP Client

○ IPv4 ○ IPv6

Protocol Type:

○ RAW TCP connection

○ Telnet (RFC2217) Server

● Telnet (RFC2217) Client

IP address : 192.168.36.12

TCP/IP port number : 9112

Use base port □

Base port number : 9100

□ Enable reconnection

Reconnection period (sec) :

Number of retries :

The Mode of Connection

If you choose the **TCP Server** mode, then the TCP/IP address of your PC will be used and you only need to specify the **TCP/IP port number** (to listen to clients' requests and communicate over the network). In this mode, **COM Port Redirector** will redirect a specified COM port to the network through a specified TCP/IP port (on your PC).

For the **TCP Server** mode, you can choose the protocol (**IPv4** or **IPv6**) the software will use with remote clients.

If you choose the **TCP Client** mode, then you must fill in the **IP address** and **TCP/IP port number** fields for a serial device or server. In this mode, **COM Port Redirector** will redirect a specified COM port on your PC to the specified TCP/IP address through a specified TCP/IP port.

In the **TCP Client** mode, the software automatically negotiates with the remote server on which protocol to use (**IPv4** or **IPv6**).

You can specify any port number between 1 and 65535 which does not cause conflicts between services and **COM Port Redirector**. To prevent conflicts between services and **COM Port Redirector**, it is recommended to use free port numbers between 1024 and 65535, i.e. unprivileged ports and not used by services. (See also the **services** system file.) A PC application employs ports on a serial server when it uses the COM ports that are created and managed by **COM Port Redirector**.

General TCP/IP settings are described in section **Using ComSetup** of chapter **Configuring VPortPro**. For more information, refer to **Appendix A. TCP/IP Connection Overview**.

With the **Use Base Port** check box disabled, you enter the absolute value for a **TCP/IP port number**. If you enable the **Use Base Port** check box, then the absolute value for a **TCP/IP port number** will be the **Base port number** plus the **COM port number**.

The Protocol Type

In the **Protocol Type** box, you must choose a transport data stream protocol type that **COM Port Redirector** will use in that connection with the serial device or server. For connection protocol, the setting must match the TCP/IP protocol that the serial server supports.

Most MS Windows applications perform only the common read/write operations that all serial devices and servers support. Some applications, however, also require some of the features provided by the COM Port Control protocol specified by IETF RFC 2217. For these applications, the serial device or server must support the COM Port Control protocol and provide at least the serial port control and status functions that the application requires.

There are two options for the protocol type between **COM Port Redirector** and the serial device or server:

RAW TCP connection

The Raw TCP connection protocol type is used to communicate with a serial device or server without any additional protocol. Although this is possible with most serial devices and servers, it is not recommended because it precludes the use of helpful Telnet (RFC 2217) protocol features.

Telnet (RFC 2217)

The Telnet protocol type is the correct setting for most serial devices and servers. If this protocol is selected, **COM Port Redirector** will automatically request the use of the Telnet "binary mode" to allow proper operation of applications. If the serial device or server supports the COM Port Control protocol (RFC 2217), it will automatically be used.

Reconnection Settings

With the **Enable reconnection** check box enabled, **COM Port Redirector** will initiate periodical attempts to reconnect the TCP/IP client connection that gets closed. If the connection fails, the connection attempt will be repeated for a user-defined number of times and interval. If the connection attempt fails beyond the user-defined limit, then a pop-up message will appear informing the user that the TCP/IP port is not available and ask if the user wants to keep retrying (the message can be disabled if desired).

This option is useful when any disruption in connection will cause **COM Port Redirector** to automatically go to stop. If the option is disabled, **COM Port Redirector** does not restart.

The **Reconnection period** setting specifies the frequency of connection attempts (in seconds). The default value is 10.

The **Number of retries** setting specifies the maximal number of connection attempts. The default value is 10. The maximal value is 1.000.000. You can enter value 0 for a very large number.

7. COM Port Redirector

< previous | content | next >

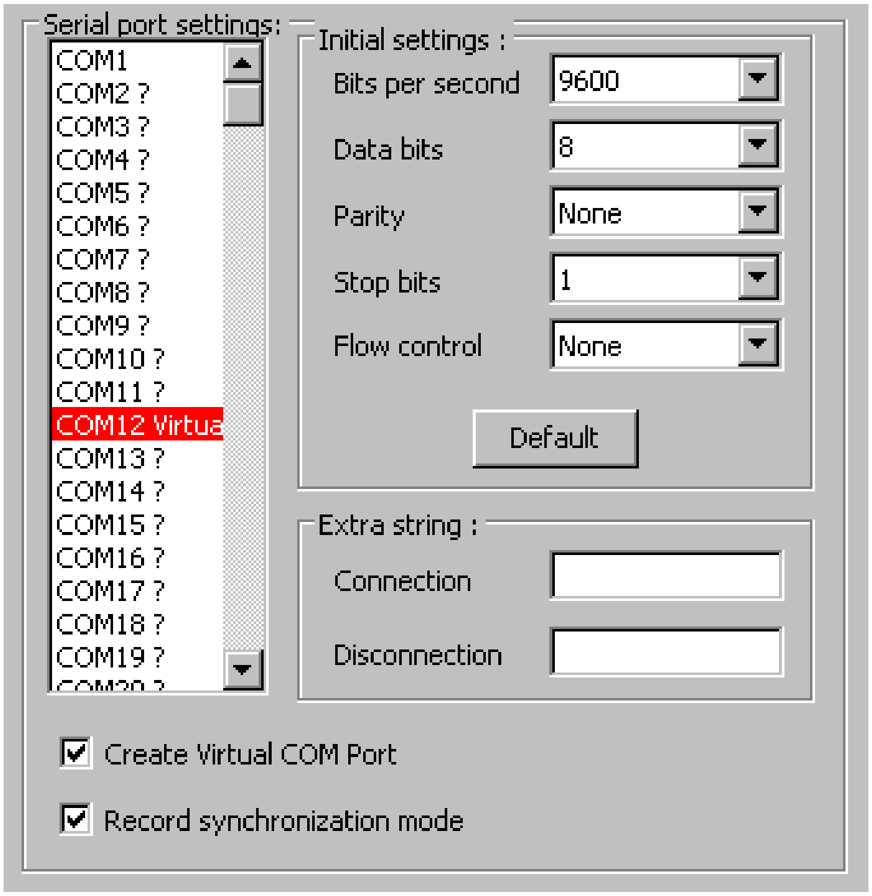
Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

< previous | content | next >

COM Port Settings

In the **Serial port settings** group box of the **Adding a Connection** or **Configure a Connection** window, you must:

1. Choose a free serial port from the list of available serial ports (when adding)
2. Make **Initial settings** for the serial port parameters (or click the **Default** button)
3. Specify a **Connection Extra string** and/or **Disconnection** one (or leave the fields empty)
4. Enable or disable the **Record synchronization mode** check box
5. Enable or disable the **Create Virtual COM Port** check box



Serial port settings:

COM1
COM2 ?
COM3 ?
COM4 ?
COM5 ?
COM6 ?
COM7 ?
COM8 ?
COM9 ?
COM10 ?
COM11 ?
COM12 Virtua
COM13 ?
COM14 ?
COM15 ?
COM16 ?
COM17 ?
COM18 ?
COM19 ?
COM20 ?

Initial settings :

Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

Extra string :

Connection:
Disconnection:

☑ Create Virtual COM Port

☑ Record synchronization mode

Create Virtual COM Port

If you disable this check box when **Adding**, then you will have to create the selected virtual COM port later before connecting (using **Configure**).
With this check box selected, the port will be created when exiting.

If you disable this check box when **Configuring**, then the selected virtual COM port will be removed from your system when exiting (with freeing its number if created). So you will have to create it later before connecting (using **Configure**). You cannot use a connection channel with a removed virtual COM port.
With this check box selected, the port will be created when exiting.

Note that COM ports listed with no sign are considered as physical ones (or created by software other than VPortPro).

Record synchronization mode

Select this check box to set the "SYNC" data transfer mode for all virtual serial ports. In this mode, data messages from/to a serial device (and serial port) are transferred unbroken in network packets. Otherwise, the "ASYNC" data transfer mode will be effective. In this mode, serial data messages may be broken into more than one network packets (i.e., into "sub-messages").

The Initial Settings Box

The **Initial settings** group box is used to change the configuration parameters for the selected COM port to their default or desired values. These are **Bits per second**, **Data bits**, **Parity**, **Stop bits**, and **Flow control**. For COM port options, the settings must match the COM port behavior expected by the PC application that will use this COM port. Note that these are initial values because they may be changed when establishing connection in the Telnet RFC-2217 protocol mode.

Default

This button is used to reset the configuration parameters for the current COM-port to their default values.

For more information, refer to **Appendix B. Serial Connection Overview**.

Bits per second

Specifies the transmission rate in bits per second. You can choose an appropriate value from the list box. The default value is 9600.

The speed at which devices transmit data is called the throughput. This is measured in bits per second (bps). Bits per second is the number of bits transmitted every second, used as a measure of the speed at which a device can transfer data. A character is made up of 8 bits. In asynchronous communication, each character may be preceded by a start bit and may terminate with a stop bit. So for each character, 10 bits are transmitted. For example, if a device communicates at 2,400 bits per second, then 240 characters are sent every second.

For serial communications, speed in bits per second can be divided by 10 to approximate the characters transmitted per second.

Data bits

Specifies the number of data bits in a character. Most systems now use 8 bits to represent a single data character (extended ASCII). In rare instances, some older systems still use 7 bits. Valid values for data bits are in the range 5 through 8. The default value is 8. Not all computers support the values 5 and 6. You can choose an appropriate number from the list box.

Parity

Specifies how the system uses the parity bit to check for transmission errors.

The parity value can be set to one of the following:

None

Send no parity bit.

Even

Set the parity bit to 0 or 1 to make the number of 1 bits even.

Odd

Set the parity bit to 0 or 1 to make the number of 1 bits odd.

Mark (1)

Set the parity bit always to 1.

Space (0)

Set the parity bit always to 0.

The default value is **None**. You can choose an appropriate value from the list box. Not all computers support the values **Mark** and **Space**. Most modem connections now use more reliable and sophisticated methods of error checking, so this is usually set to **None**.

Stop bits

Specifies the number of stop bits that define the end of a character: 1, 1.5, or 2. You can choose an appropriate number from the list box. The default value is 1. If the baud rate is 110, the default value is 2; otherwise, the default value is 1. Not all computers support the value 1.5.

Flow control

From within the list box, you can choose one of the following:

XON / XOFF

Specifies whether the XON or XOFF protocol for data-flow control is on or off.

Software flow control (also called XON/XOFF or CTRL+S/CTRL+Q) uses data characters to indicate that the data flow should start or stop. This enables a device to send a control character to signal another device to stop transmitting while it catches up.

Software flow control is slower and usually less desirable than hardware flow control. Software flow control is used only for transmitting text. It cannot be used for binary file transfer because binary data may contain the special flow control characters.

Hardware

Hardware flow control (RTS/CTS) depends on the device to control the flow of data. This should be used with all high-speed modems or modems that compress data.

RTS specifies whether the Request To Send circuit is set to on, off, handshake, or toggle.

CTS specifies whether output handshaking that uses the Clear To Send circuit is on or off.

None

No flow control options will be used (the default value).

Note

When you want to "connect" a serial data source with a serial data destination (e.g., across the network or as two virtual serial ports), make sure that the serial port settings on both sides of such a "connection" be matched.

The Extra String Box

In the **Extra String** box, you can specify the following edit fields:

Connection

When the TCP/IP connection gets established, the bytes sequence you specified in the field will first be sent to the COM port.

Disconnection

When the TCP/IP connection gets closed, the bytes sequence you specified in the field will finally be sent to the COM port.

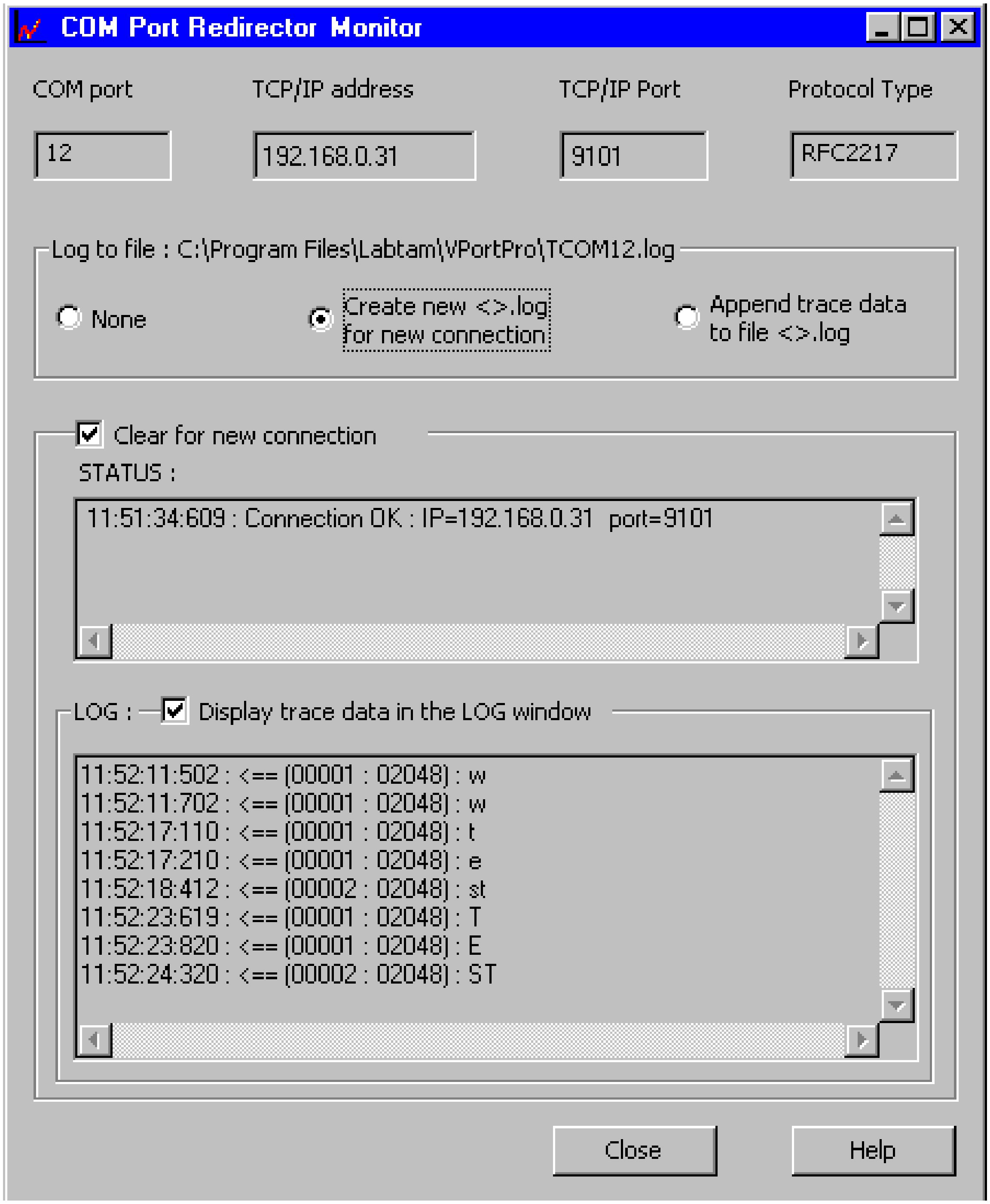
These options may be useful if the TCP/IP client is suddenly disconnected from the TCP/IP server while attached via **COM Port Redirector** to one of remote sites. To disconnect the modem from the remote site on the unattended server, this is a way to be able to drop DTR, etc.

7. COM Port Redirector < previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

Monitoring COM Ports

From the **COM Port Redirector** window, you can select a connection channel and press the **Port monitor** button to display the **COM Port Redirector Monitor** window. Then you can watch activity of a selected COM port and control data logging for that channel.



COM port	TCP/IP address	TCP/IP Port	Protocol Type
12	192.168.0.31	9101	RFC2217

Log to file : C:\Program Files\Labtam\VPortPro\TCOM12.log

○ None ● Create new <>.log for new connection ○ Append trace data to file <>.log

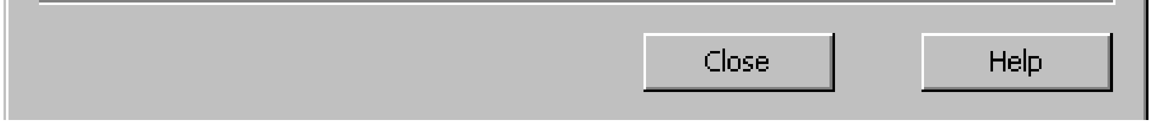
☑ Clear for new connection

STATUS :

```
11:51:34:609 : Connection OK : IP=192.168.0.31  port=9101
```

LOG : ☑ Display trace data in the LOG window

```
11:52:11:502 : <== (00001 : 02048) : w
11:52:11:702 : <== (00001 : 02048) : w
11:52:17:110 : <== (00001 : 02048) : t
11:52:17:210 : <== (00001 : 02048) : e
11:52:18:412 : <== (00002 : 02048) : st
11:52:23:619 : <== (00001 : 02048) : T
11:52:23:820 : <== (00001 : 02048) : E
11:52:24:320 : <== (00002 : 02048) : ST
```



COM port displays a serial port number for the connection channel.

TCP/IP address displays a TCP/IP address of a server for the connection channel.

TCP/IP Port displays a TCP/IP port of the server for the connection channel.

Protocol Type displays a protocol type used for the connection channel.

Close closes the **COM Port Redirector Monitor** window and the COM port trace log-file (if specified).

The Log to file Box

By using the following radio buttons, you can choose one of trace options you prefer:

None

Specifies that no log-data will be stored in the log-file shown.

Create new <>.log for new connection

Specifies to create a new log-file with the name shown to store new log-information for the connection channel seen under the header of the dialog box. The log-file name reflects the serial port number (as a suffix).

Append trace data to file <>.log

Specifies to append new log-information to the existing log-file (with the name shown) for the connection channel seen under the header of the dialog box. The log-file name reflects the serial port number (as a suffix).

The Clear for New Connection Box

When enabled, **Clear for new connection** cleans up the STATUS and LOG fields when a new connection is established.

STATUS

In this field, information about the status of the connection channel is displayed.

Display trace data in the LOG window

With this check box enabled, log-information for the connection channel will be displayed in the LOG window below.

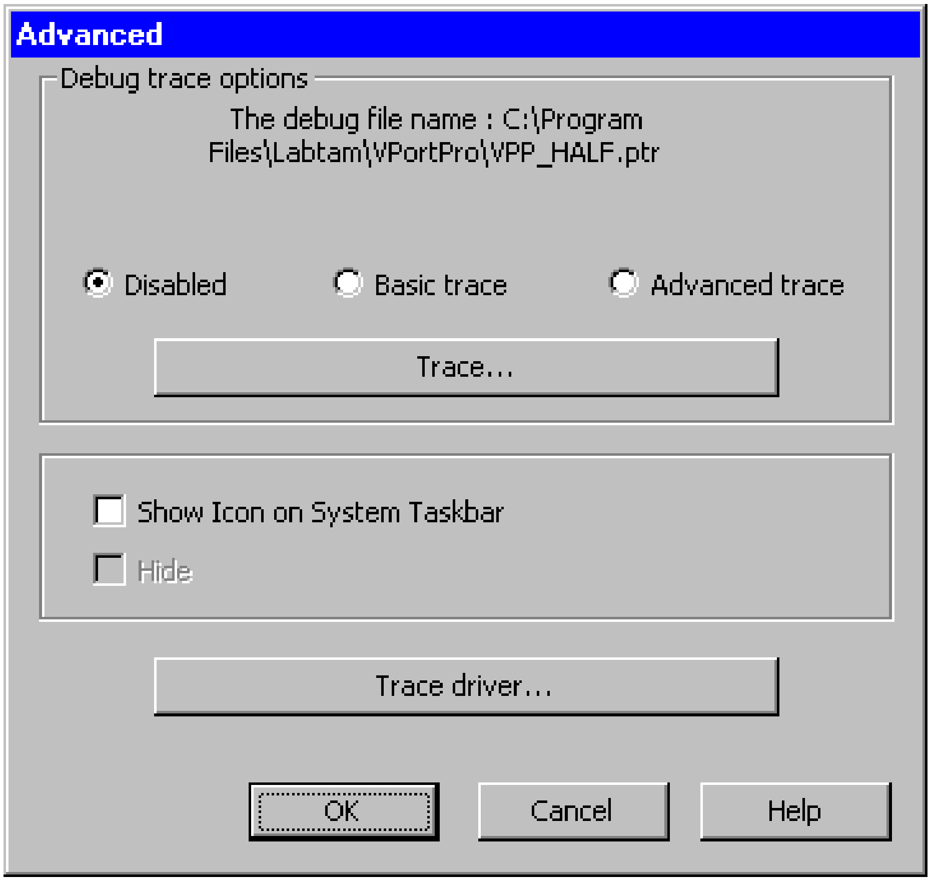
< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

< **previous** | **content** | **next** >

The Advanced Box

Clicking the **Advanced** button in the **COM Port Redirector** window will come up with the **Advanced** dialog box that lets you specify your preferences for all connection channels in the list:



Advanced

Debug trace options

The debug file name : C:\Program Files\Labtam\VPortPro\VPP_HALF.ptr

● Disabled ○ Basic trace ○ Advanced trace

□ Show Icon on System Taskbar

□ Hide

By using the following check boxes, you can choose whether the **COM Port Redirector** window will be on the desktop or hidden on the system taskbar:

Show Icon on System Taskbar

Specifies to place the **COM Port Redirector** icon on the system taskbar. Using this icon, you can **Exit** the program, or **Configure** it (i.e. to restore displaying the **COM Port Redirector** window).

Hide

With the **Show Icon on System Taskbar** check box enabled, specifies to hide the **COM Port Redirector** window from the desktop.

Clicking the **Trace driver** button calls the tracing utility (described in section **Trace Settings**).

The Debug Trace Options Box

By using the following radio buttons on the **Debug trace options** box, you can choose one of trace options you prefer:

Disabled

Specifies that no log-data will be stored in the log-file shown (and displayed in the LOG window).

Basic trace

Specifies to create a new log-file with the name shown to store brief log-information for the connection channel (and displayed in the LOG window).

Advanced trace

Specifies to create a new log-file with the name shown to store full log-information for the connection channel (and displayed in the LOG window).

Trace

Displays the **Trace** window for you to watch activity of active COM ports (see section **Tracing Active COM ports**).

< previous | content | next >

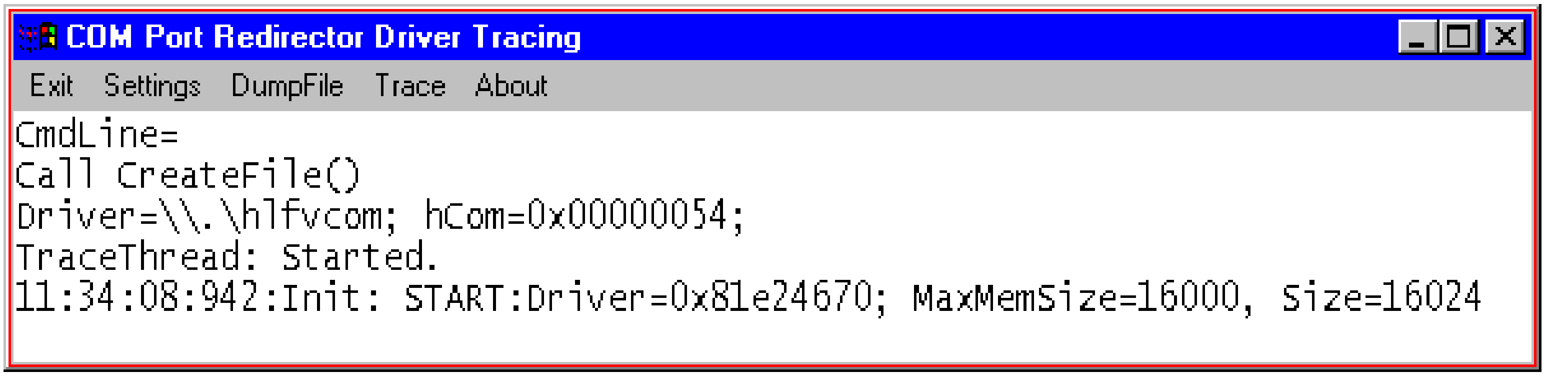
Trace Settings

The tracing utility is designed for setting a number of trace flags (by toggling related check boxes to filter out data of your interest) and passing them to the virtual serial port driver. According to the Trace Flags set, utilities of the package will filter out and save related information into corresponding trace files you can analyze later. The tracing could be used to document the virtual serial port traffic between your software applications that communicate with serial servers via connection channels.

Normally, users should not arbitrarily turn on/off Trace Flags. The flags can noticeably slow down the traffic and can generate a great amount of data. The tracing utility is primarily meant for testing and analyzing how your virtual serial port software applications interface to access servers while using serial port communication features.

Driver Tracing Settings

You can start the tracing program by clicking the **Trace driver** button in the **Advanced** dialog box. The **COM Port Redirector Driver Tracing** window will appear on your display:



```
CmdLine=
Call CreateFile()
Driver=\\.\hlfvcom; hCom=0x00000054;
TraceThread: Started.
11:34:08:942:Init: START:Driver=0x81e24670; MaxMemSize=16000, Size=16024
```

In this window, the program displays event tracing information according to trace flags you specified for the virtual serial port driver in the **Trace Settings** dialog box described below. The information displayed will also be saved in a log file (if specified in the **Trace Settings** dialog box).

Pressing **Exit** on the menu bar closes the window and exits the program.

Pressing **Settings** on the menu bar brings up the **Trace Settings** dialog box. (Refer to section **Making Trace Settings** below.)

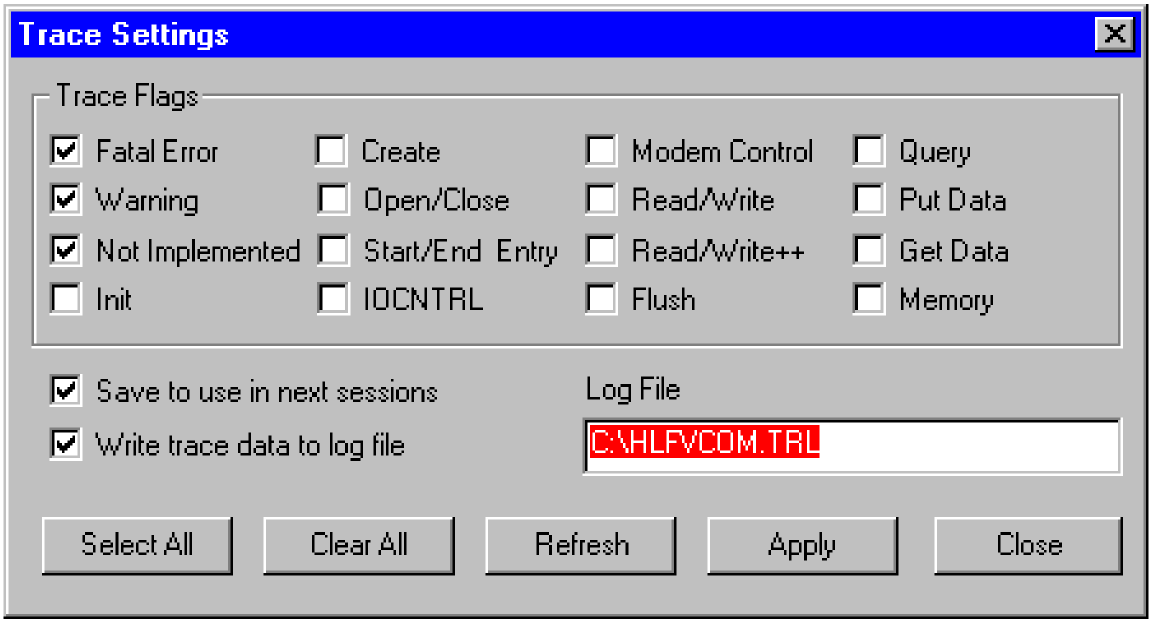
Pressing **DumpFile** on the menu bar brings up the **Dump File** dialog box. (Refer to section **The**

Dump File Box below.)

Pressing **Trace** on the menu bar brings up a submenu where you can choose to start or stop tracing.

Making Trace Settings

Pressing **Settings** on the menu bar of the **COM Port Redirector Driver Tracing** window brings up the **Trace Settings** dialog box:



☑ Fatal Error	□ Create	□ Modem Control	□ Query
☑ Warning	□ Open/Close	□ Read/Write	□ Put Data
☑ Not Implemented	□ Start/End Entry	□ Read/Write++	□ Get Data
□ Init	□ IOCNTRL	□ Flush	□ Memory

☑ Save to use in next sessions

☑ Write trace data to log file

Log File: C:\HLFVCOM.TRL

If you click the **Close** button, the program closes the dialog with no changes.

Pressing the **Apply** button passes the Trace Flags set to the virtual serial port driver. The Trace Flags will take effect immediately.

Select All makes all the Trace Flags enabled. **Clear All** makes all the Trace Flags disabled.

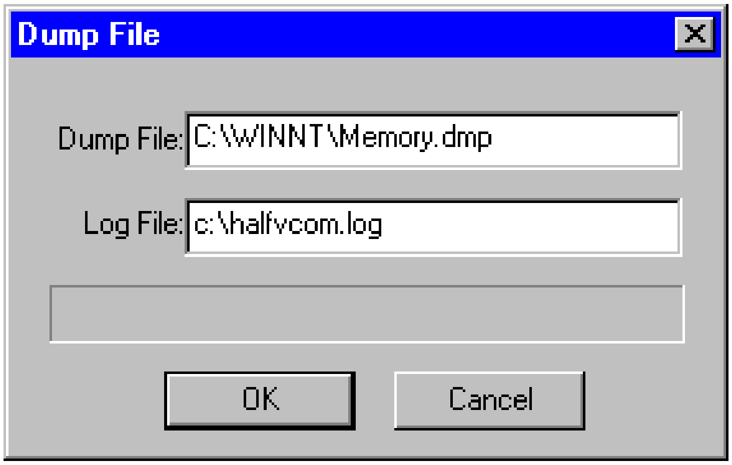
Refresh restores all Trace Flags to their initial states (when loaded the tracing utility or after the latest **Apply**).

With the **Save to use in next sessions** check box enabled, you can store the current settings (by pressing **Apply**) and load them later (when loading the tracing utility).

With the **Write trace data to log file** check box enabled, you can enter a full path file name or leave the default one in the **Log File** field to store flagged log information into that file.

The DumpFile Box

Pressing **DumpFile** on the menu bar of the **COM Port Redirector Driver Tracing** window brings up the dialog box:



Dump File: C:\WINNT\Memory.dmp

Log File: c:\halfvcom.log

You can specify a full path name for one of system dump files to be read (when system crash) and a log file name to store the driver tracing information extracted from that dump file that you can analyze later. Normally, you can use the default file names shown.

< previous | content | next >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

The "[VPORTPRO]" Section of the ini-file

The "[VPORTPRO]" section of the ini-file may have the following entry lines you can customize for particular needs and applications.

DCD=L

Data Carrier Detect (DCD) - name of the serial communication signal (defined by the RS-232-C specification) that indicates the presence or absence of a basic carrier signal from a (remote) DCE (Data Communications Equipment). With the help of the DCD line DCE reports to DTE (Data Terminal Equipment) that the connection with the remote device is established and ready for transmission.

The **L** value in this entry line specifies the way the software will interpret the DCD "state" of virtual serial ports (e.g., to ensure that client and server parts are properly connected to each other across the network by virtual "serial cable" when you plug in a serial device).

- **DCD=0** - DCD is constantly **Off**
- **DCD=1** - DCD is constantly **On** (Default)
- **DCD=2** - DCD is controlled by the DTR signal (as in NULL Modem Cable).
  This is applicable only for the **RFC2217** protocol type (not for **RAW**).
- **DCD=4** - DCD will be turned **On** when client and server connects to each other,
  and DCD will be turned **Off** when client and server disconnects from each other.

Note that all these values are applicable for both the **RFC2217** and **RAW** protocol types except for **DCD=2**.

SYNC=L

The **SYNC=1** entry line specifies the "SYNC" data transfer mode for all virtual serial ports. In this mode, data messages from/to a (virtual) serial device (and serial port) are transferred unbroken in network packets.

With **SYNC=0**, the "ASYNC" data transfer mode will be effective. In this mode, serial data messages may be broken into more than one network packets (i.e., into "sub-messages").

< previous | content | next >

Using as Service

Using COM Port Redirector Service

This section describes how to start and use Com Port Redirector as a standard MS Windows service.

The functionality and settings of **Com Port Redirector Service** are just the same as of Com Port Redirector itself. Additionally, you can control the behavior of Com Port Redirector as a MS Windows service.

To manage the session of Com Port Redirector Service (e.g., to start/stop services as administrator), you can use Control Panel/Administrative Tools/Services of MS Windows.

Services overview

A service is an application type that runs in the background and is similar to UNIX daemon applications. Service applications typically provide features such as client/server applications, Web servers, database servers, and other server-based applications to users, both locally and across the network.

You can use Services to:

- Start, stop, pause, resume, or disable services on remote and local computers (including remote computers running Windows NT 4.0.). You must have the appropriate permissions to start, stop, pause, restart, and disable services.
- Manage services on local and remote computers (on remote computers running Windows XP, Windows 2000, or Windows NT 4.0 only).
- Create custom names and descriptions for services so that you can easily identify them (on computers running Windows XP or Windows 2000 only).
- Configure startup options.
- Set up recovery actions to take place if a service fails, for example, restarting the service automatically or restarting the computer (on computers running Windows XP or Windows 2000 only).
- Enable or disable services for a particular hardware profile.
- View the status and description of each service.

Services permissions

Each service has special permissions that you can grant or deny for each user or group. You can set permissions for individual services by using Security Templates.

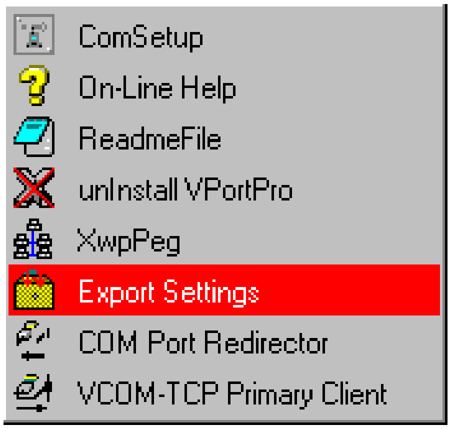
According to MS Windows Help Manual, Services must log on to an account in order to access resources and objects on the operating system. Some services are configured by default to log on to the **Local System account**, which is a powerful account that has full access to the system. If a service logs on to the **Local System account** on a domain controller, that service has access to the entire domain. Other services are configured to log on to **LocalService** or **NetworkService** accounts, which are special built-in accounts that are similar to authenticated user accounts. These accounts have the same level of access to resources and objects as members of the **Users** groups. This limited access helps safeguard your system if individual services or processes are compromised.

Services running as the **LocalService** account access network resources as a null session with no credentials. Services running as the **NetworkService** account access network resources using the credentials of the machine account.

< previous | content | next >

8. Exporting Settings

By clicking on the **Export Settings** item in the VPortPro Programs' folder (i.e., from the Start/Programs/VPortPro menu):



you can store all the settings you have made on the COM-TCP/IP connection channel list in the **VCOM-TCP Primary Client** and **COM Port Redirector** windows into the **vppini.dat** file. This VPortPro configuration file will be created in the home directory and include data in the ASCII format.

With **Export Settings**, you can create several VPortPro configuration files with different settings you need (do not forget to rename the **vppini.dat** file before next exporting).

The VPortPro configuration file, **vppini.dat**, is used during secondary installations of the VPortPro package (See section **The Silent Installation Mode** in chapter **Running Setup**). If you copy the file into the distributive directory, Setup will install the package and restore configuration files from this file (thus exporting settings from one installation site to another one).

< previous | content | next >

< previous | content | next >

Appendix A. TCP/IP Connection Overview

The TCP/IP protocol is designed such that each PC or device on a network has a unique "IP Address" (Internet Protocol Address) and each IP address can open and communicate over up to 65535 different "ports" for sending and receiving data to or from any other network device. The IP Address identifies the PC or device and a "Port Number" identifies a specific connection between one PC or device and another. A TCP/IP "port" can be thought of as a private communications line where the port number is used to uniquely identify each unique connection between two devices. The concept is very similar to any other type of port on your PC (serial, parallel, etc) except that instead of having a physical connection, the TCP/IP protocol creates a "virtual port" and the network software is responsible for routing the data in and out of each virtual port.

Most computer operating systems that support networking are now shipped with complete support for the TCP/IP protocol as well as tools for performing simple TCP/IP I/O operations. For example, MS Windows comes with a program called "Telnet" that is similar to a traditional "Terminal" program except that it is designed for TCP/IP communications instead of serial I/O. The HyperTerminal program that is shipped with MS Windows also supports TCP/IP communications.

Each computer system is known as a host (also known as nodes in other networking lexicons) A system that performs a task for a remote host is called a server; the host for whom the system is performed is called the client.

The IP address is a 32-bit address used to identify a node on an IP network. Each node on the IP network must be assigned a unique IP address, which is made up of a network identifier and a host identifier. This address is typically represented in dotted-decimal notation, with the decimal value of each octet separated by a period, for example, 192.168.9.32.

The PC that **VPortPro** software is running on must have at least one network interface card with an IP address assigned to it. In MS Windows, the TCP/IP protocol can be configured to automatically obtain an IP address from a host computer. This means that your PC may not have an IP address until it is connected to a network server or a host computer. You may need to contact your network administrator to assign an IP address to your PC if you wish to configure a COM port using a TCP/IP server connection. This is done in the network settings for the TCP/IP protocol in your control panel.

The IP address that you specify when configuring **VPortPro** may also be either a URL or the name of a computer located on your network. If you are configuring **VPortPro** and your network is set up to assign IP addresses dynamically to each individual workstation, then you may need to use the name of the PC that you want to connect to instead of an actual IP address in order to guarantee a connection.

When you configure **VPortPro** you will need to specify the IP address and the port number of the remote TCP/IP server that you want to connect to. General TCP/IP settings are described in section **Using ComSetup** of chapter **Configuring VPortPro**.

Notes on Telnet protocol and RFC 2217

There are three new areas of functionality (added to the Telnet protocol and described in RFC 2217) to successfully support the needs of outbound modem dialing (i.e., transmitting data from the modem attached to the access server to a remote service). These are:

- The ability for the client (i.e., any network device which initiates a Telnet session to an access server) to send serial port configuration information to the access server (i.e., any network device which accepts Telnet sessions and passes the data received to a serial port, and passes data received from the serial port to the client via the Telnet session) that is connected to the outbound modem. This is needed to ensure the data being transmitted and received by the modem is formatted correctly at the byte level.
- The ability for the access server to inform the client of any modem line or signal changes such as RLSD changes (carrier detect). This information is vital, since many client software packages use this information to determine if a session with the remote service (i.e., any service which accepts dial-up connections) has been established. RLSD changes are also used for signaling in Class I faxing.
- The ability to manage flow control between the client and the access server which does not interfere with the flow control mechanisms used by the session between the client and the remote service. Unfortunately, RFC 1372 "Telnet Remote Flow Control Option" cannot be used for this purpose because it relies on sending XON/XOFF style characters which maybe transmitted or received as a normal course of the client/remote service session.

< **previous** | **content** | **next** >

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us

< previous | content

Appendix B. Serial Connection Overview

Notes on the RS232 Standard

There are two basic types of serial communications, synchronous and asynchronous. With synchronous communications, the two devices initially synchronize themselves to each other, and then continually send characters to stay in sync. Even when data is not really being sent, a constant flow of bits allows each device to know where the other is at any given time. That is, each character that is sent is either actual data or an idle character. Synchronous communications allows faster data transfer rates than asynchronous methods, because additional bits to mark the beginning and end of each data byte are not required. The serial ports on IBM-style PCs are asynchronous devices and therefore only support asynchronous serial communications.

Asynchronous means "no synchronization", and thus does not require sending and receiving idle characters. However, the beginning and end of each byte of data must be identified by start and stop bits. The start bit indicates when the data byte is about to begin and the stop bit signals when it ends. The requirement to send these additional two bits cause asynchronous communications to be slightly slower than synchronous however it has the advantage that the processor does not have to deal with the additional idle characters.

An asynchronous line that is idle is identified with a value of 1 (also called a mark state). By using this value to indicate that no data is currently being sent, the devices are able to distinguish between an idle state and a disconnected line. When a character is about to be transmitted, a start bit is sent. A start bit has a value of 0 (also called a space state). Thus, when the line switches from a value of 1 to a value of 0, the receiver is alerted that a data character is about to come down the line.

The RS232 port was intended to be a general input/output port, thereby allowing data to be sent and received to and from many different types of external devices. The standard defines the electrical characteristics (i.e. connector pin-outs, voltage levels and electrical signals) for transmitting data from one electronic device to another. The RS232 interface is also called a "serial" interface because digital data is transmitted "serially" one bit at a time. Bytes of data are taken apart by a transmitter with each bit being transmitted in series. The receiving device collects the bits and puts them back together into the original data bytes. The content and structure of any data that is transmitted is not defined by the RS232 standard, however, the most commonly used format for data is text and decimal numbers, i.e. readable data (at least in the measuring instrument industry).

In addition to RS232, RS485 and RS422 data is also serial. RS 485 is also called "multi-drop" as you can address multiple devices over an RS485 cable. Both RS485 and RS422 are easily converted to RS232 or you can purchase special ports for your PC.

Every PC in the world comes equipped with at least one or two serial ports (RS232 connectors). Even laptops and palmtops come with at least one serial port. All IBM PC and compatible computers are typically equipped with two serial ports and one parallel port. A serial port sends and receives data one bit at a time over one wire. Two-way (full duplex) communications is possible with only three separate wires - one to send, one to receive, and a common signal ground wire. The serial port on your PC is a full-duplex device meaning that it can send and receive data at the same time. In order to be able to do this, it uses separate lines for transmitting and receiving data.

Stop bits frame data packets in asynchronous communication. These tell the receiving device that a byte has been sent. Because data is received at irregular intervals, the receiving device must be signaled to let it know when the data bits of a character begin and end. This is done by means of start and stop bits. Modern asynchronous protocols never require more than one stop bit.

Once the start bit has been sent, the transmitter sends the actual data bits. There may either be 5, 6, 7, or 8 data bits, depending on the number you have selected. Both receiver and the transmitter must agree on the number of data bits, as well as the baud rate. Almost all devices transmit data using either 7 or 8 data bits.

Notice that when only 7 data bits are employed, you cannot send ASCII values greater than 127. Likewise, using 5 bits limits the highest possible value to 31.

After the data has been transmitted, a stop bit is sent. A stop bit has a value of 1 - or a mark state - and it can be detected correctly even if the previous data bit also had a value of 1. This is accomplished by the stop bit's duration. Stop bits can be 1, 1.5, or 2 bit periods in length. Note that stop bit 1.5 is supported by most serial port hardware only if data size is set to 5 bits. It is not commonly used.

The transmission rate (or baud rate) is the speed at which a device communicates. Baud rate refers to the number of times the condition of the line changes. This is equal to bits per second only if each signal corresponds to one bit of transmitted data.

Modems must operate at the same baud rate in order to communicate with each other. If the baud rate of one modem is set higher than that of the other, the faster modem usually alters its baud rate to match that of the slower modem.

Besides the synchronization provided by the use of start and stop bits, an additional bit called a parity bit may optionally be transmitted along with the data. A parity bit affords a small amount of error checking, to help detect data corruption that might occur during transmission. You can choose even parity, odd parity, mark parity, space parity or none at all. When even or odd parity is being used, the number of marks (logical 1 bits) in each data byte is counted, and a single bit is transmitted following the data bits to indicate whether the number of 1 bits just sent is even or odd.

For example, when even parity is chosen, the parity bit is transmitted with a value of 0 if the number of preceding marks is an even number. For the binary value of 0110 0011 the parity bit would be 0. If even parity were in effect and the binary number 1101 0110 were sent, then the parity bit would be 1. Odd parity is just the opposite, and the parity bit is 0 when the number of mark bits in the preceding word is an odd number.

Parity error checking is very rudimentary. While it will tell you if there is a single bit error in the character, it doesn't show which bit was received in error. Also, if an even number of bits were in error then the parity bit would not reflect any error at all.

Mark parity means that the parity bit is always set to the mark signal condition and likewise space parity always sends the parity bit in the space signal condition. Since these two parity options serve no useful purpose whatsoever, they are almost never used.

Flow control is the method by which a device controls the rate at which other devices send data to it. There are six different links involved in a connection: transmitting computer to transmitting modem, transmitting modem to receiving modem, receiving modem to computer, and the reverse of all three links. They can all have different data transmission rates. When the receiving modem is temporarily unable to accept the data, it needs a way to tell the transmitting modem to slow down or wait for it.

If flow control is improperly configured, it may be impossible to connect to a remote system, the transfer rate may be considerably slowed, or the connection may be disrupted. If you see many errors and data retransmissions, check both the device and the communication program settings for flow control. Flow control settings for both the communication program and the device must be the same. Many communication programs set these automatically based on the device, but some must be configured separately.

Notes on DTE and DCE

Two terms you should be familiar with are DTE and DCE. DTE stands for Data Terminal Equipment, and DCE stands for Data Communications Equipment. These terms are used to indicate the pin-out for the connectors on a device and the direction of the signals on the pins. Your computer is a DTE device, while most other devices are usually DCE devices.

If you have trouble keeping the two straight then replace the term "DTE device" with "your PC" and the term "DCE device" with "remote device" in the following discussion.

The RS-232 standard states that DTE devices use a 25-pin male connector, and DCE devices use a 25-pin female connector. You can therefore connect a DTE device to a DCE using a straight pin-for-pin connection. However, to connect two like devices, you must instead use a null modem cable. Null modem cables cross the transmit and receive lines in the cable

RTS stands for Request To Send. This line and the CTS line are used when "hardware flow control" is enabled in both the DTE and DCE devices. The DTE device puts this line in a mark condition to tell the remote device that it is ready and able to receive data. If the DTE device is not able to receive data (typically because its receive buffer is almost full), it will put this line in the space condition as a signal to the DCE to stop sending data. When the DTE device is ready to receive more data (i.e. after data has been removed from its receive buffer), it will place this line back in the mark condition. The complement of the RTS wire is CTS, which stands for Clear To Send. The DCE device puts this line in a mark condition to tell the DTE device that it is ready to receive the data.

Likewise, if the DCE device is unable to receive data, it will place this line in the space condition. Together, these two lines make up what is called RTS/CTS or "hardware" flow control. COM Port Redirector supports this type of flow control, as well as Xon/Xoff or "software" flow control. Software flow control uses special control characters transmitted from one device to another to tell the other device to stop or start sending data. With software flow control the RTS and CTS lines are not used.

DTR stands for Data Terminal Ready. Its intended function is very similar to the RTS line. DSR (Data Set Ready) is the companion to DTR in the same way that CTS is to RTS. Some serial devices use DTR and DSR as signals to simply confirm that a device is connected and is turned on. COM Port Redirector sets DTR to the mark state when the serial port is opened and leaves it in that state until the port is closed. The DTR and DSR lines were originally designed to provide an alternate method of hardware handshaking. It would be pointless to use both RTS/CTS and DTR/DSR for flow control signals at the same time. Because of this, DTR and DSR are rarely used for flow control.

< **previous** | **content**

Home | Product | Download | Order Now | Upgrade | Support | Pricing | Company Information | Contact Us