

ReMapPro V2.9 Online Help manual

1. About This Manual
 2. Introducing to ReMapPro
 - 2.1. What is in ReMapPro
 3. The ReMapPro Requirements
 4. Installing ReMapPro
 - 4.1. Running Setup
 - 4.2. Running UNINSTALL
 - 4.3. Upgrading ReMapPro
 - 4.4. Multi-user Installation
 5. Configuring ReMapPro
 - 5.1. Using ComSetup
 - 5.2. Running ComSetup with Command Line Parameters
 - 5.3. The "[NETWORK]" Section of the ini-file
 6. Comport
 - 6.1. Using Comport
 - 6.2. Virtual Serial Ports
 - 6.3. Configuring Comport
 - 6.4. Monitoring COM Ports
 - 6.5. Exporting Settings
 - 6.6. The "[COMPORT]" Section of the ini-file
 7. Comserv
 - 7.1. Installing Comserv
 - 7.2. Configuring Comserv
 - 7.3. Example of running COMSERV as a MS Windows 2000/XP service
 - 7.4. The "[COMSERV]" Section of the ini-file
 8. DuplexDr
 - 8.1. What is in DuplexDr
 - 8.2. Using DuplexDr
 - 8.3. Configuring DuplexDr
 - 8.4. Installing DuplexDr
 - 8.5. Uninstalling DuplexDr
 - 8.6. Tracing DuplexDr
- Appendix A. TCP/IP Connection Overview
- Appendix B. Serial Connection Overview

[content](#) | [next](#) >

1. About This Manual

This User's Manual describes how to install, configure and use the ReMapPro package on a 32-bit IBM PC or compatible personal computer running one of the following operating systems: MS Windows 9x/ME/NT4/2K/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 ReMapPro
- Hardware & Software requirements of ReMapPro
- Installation procedure for ReMapPro
- Configuring ReMapPro
- ReMapPro working sessions.

[content](#) | [next](#) >

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)

[< previous](#) | [content](#) | [next >](#)

2. Introducing to ReMapPro

The main utility, Comport, included in the ReMapPro package makes serial (RS232) data from your PC available on TCP/IP-based networks and makes TCP/IP data available on serial ports of your PC. This makes it possible for any other computer on the same network to send and receive serial data through a remote serial port on the PC where Comport is running. You can connect to the remote serial port by connecting to a TCP/IP port. Comport can also be used to pass serial data across a corporate intranet or over the Internet. Comport allows multiple sessions to run simultaneously but each with different serial port.

Comport supports 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.

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)

2. Introducing to ReMapPro

[< previous](#) | [content](#) | [next >](#)

What is in ReMapPro

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

Comport

Comport 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, Comport can receive serial data from a COM port of your PC and then send them out to a TCP/IP network through a TCP/IP socket connection on your PC. Comport 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.

While you are using Comport, you can:

- Specify, save and load COM-port settings and associated TCP/IP port settings (i.e. configure COM-TCP/IP connection channels)
- Start Comport as a TCP/IP client (with one-to-one mapping of COM and IP ports)
- Start Comport as a TCP/IP server (with one-to-one mapping of IP and COM ports)
- Start multiple Comport sessions with different connection channels settings
- Start Comport as a TCP/IP server with multiple COM ports on a single IP port (so that multiple modems could be used on the same IP port with auto-switching from one modem to the next)
- Control running Comport sessions (start/stop/pause/resume connection channels, view connection channels statistics, monitor data logging for a specified serial port).

Also with Comport, you can create "virtual COM ports" and use "the virtual serial ports pool" in your system. 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. The VSP pool can be used to supply any software application with COM ports (e.g., this is vitally necessary on some modern notebooks that do not have physical COM ports at all). This feature is available after installation of the VSP driver included in the package. With the VSP driver installed, you can configure Comport 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.

DuplexDr

With DuplexDr, you can create a pair of virtual serial ports (VSP pair) "virtually connected to each other" in your system. A virtual serial port appears to application software as a "standard" MS Windows'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. Entering VSP pairs, you can create and use "a VSP pairs pool". The pool of VSP pairs can be used to supply any MS Windows application with serial ports as though they were standard serial ports and physically connected in pairs. This feature is available after installation of the corresponding driver included in the package. This allows MS Windows applications to generate or consume data to and from VSP as though the data were being processed by physical hardware.

DuplexDr can be used for sending and receiving data between two separate MS Windows applications running on your PC, thus connecting one application to another by using a VSP pair (for example, Comport and HyperTerminal). Also, you can use the tracing utility, XVCTRC, to document the virtual serial port traffic between your software applications that communicate with such a VSP pair channel (e.g., for application development testing).

Comserv

Comserv is a software program that designed to be used as a standard MS Windows NT service. Comserv's functionality and settings are just the same as of Comport. Additionally, you can control the behavior of Comserv as the NT service.

ComSetup

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

2. Introducing to ReMapPro

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)

[< previous](#) | [content](#) | [next >](#)

3. The ReMapPro Requirements

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

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
- 11 Mbytes free hard disk space.

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

- MS Windows 9x/ME/NT4/2K/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 ReMapPro

This chapter describes how to install the ReMapPro software. The chapter assumes that you have one of the Microsoft Windows 9x/ME/NT4/2K/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 ReMapPro 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 >](#)

4. Installing ReMapPro

[< 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 to your hard disk
2. Execute the file and select **OK**.

In the dialog box that appears, 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 the distributive directory shown in the field.

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 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:

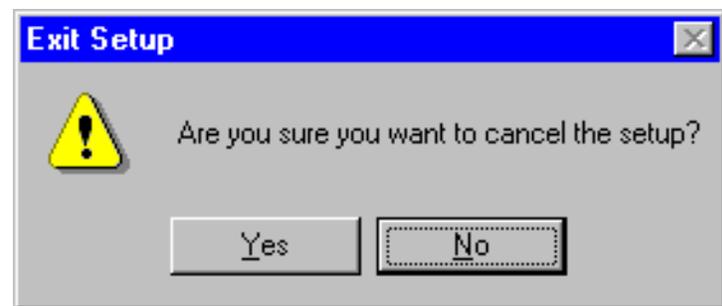


The installation procedure (i.e. running Setup) will start automatically from the temporary distributive directory if you enable the **When done unzipping open: .\setup.exe** check box.

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.



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).

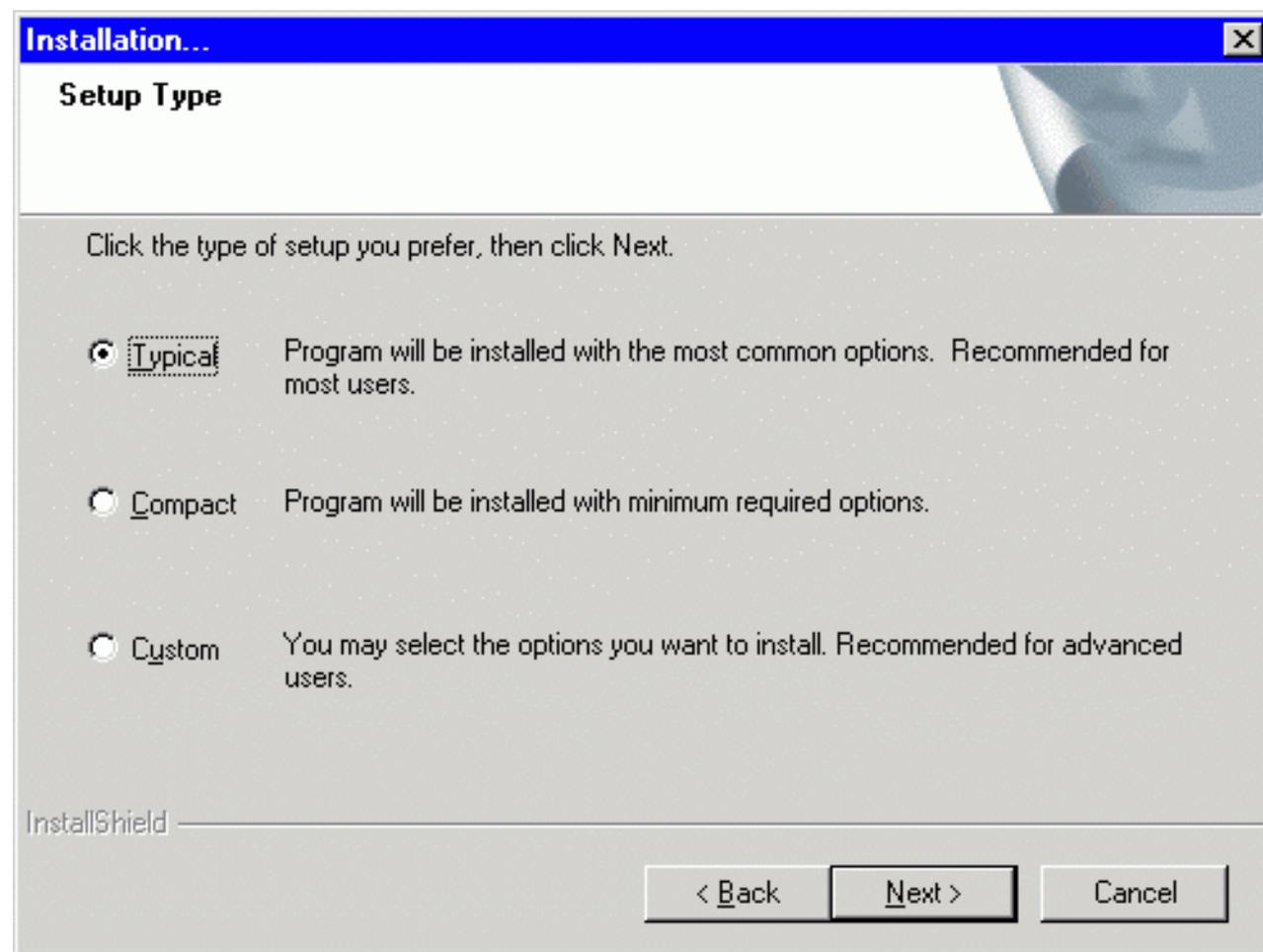
- **Choose Destination Location**

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.

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 ReMapPro** below.)

- **Setup Type**

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



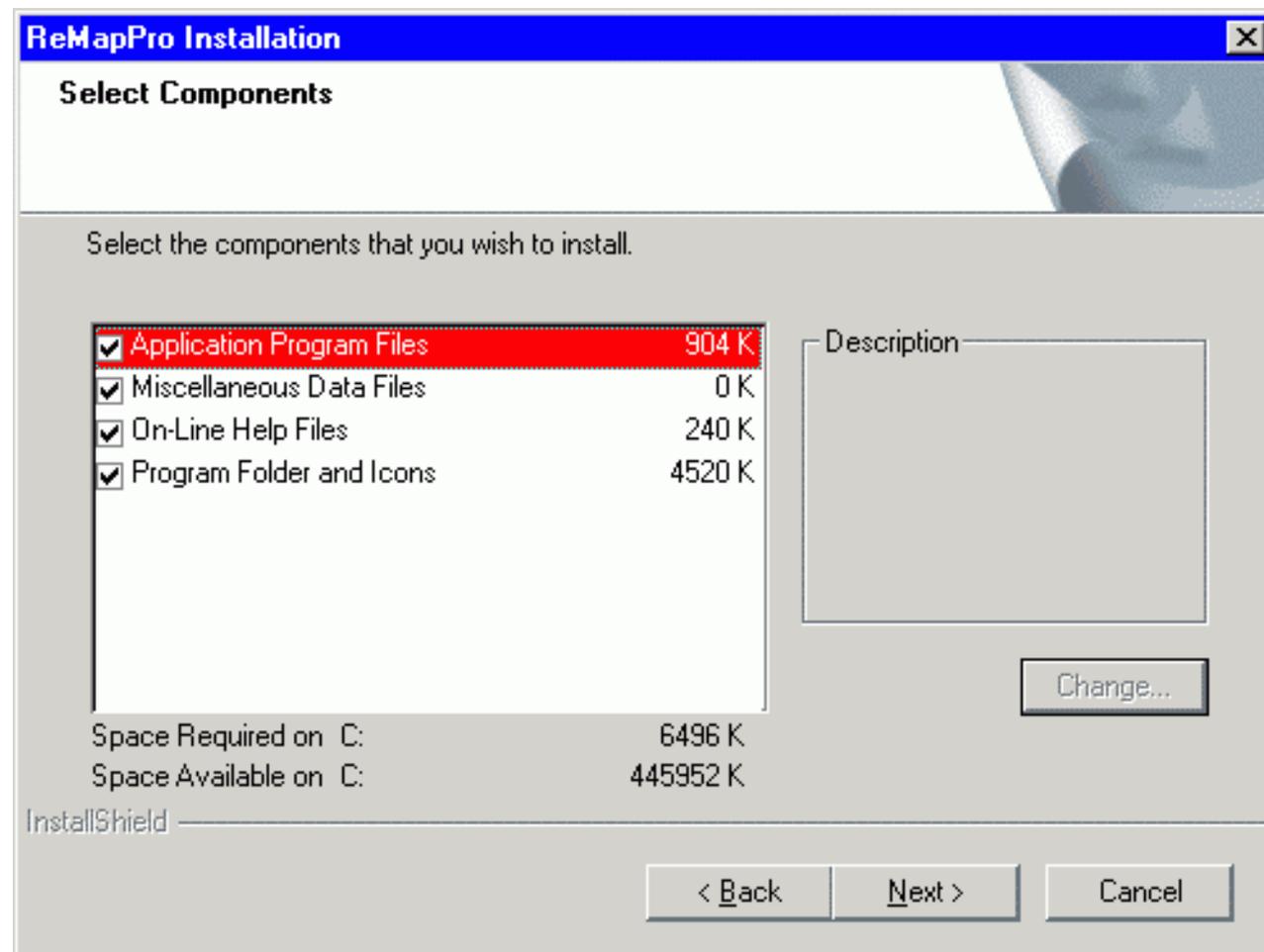
There are three Setup types: **Typical**, **Compact**, and **Custom**.

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).



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.

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 ReMapPro Program folder with shortcuts for the package components. It also adds the ReMapPro item to the Programs menu.

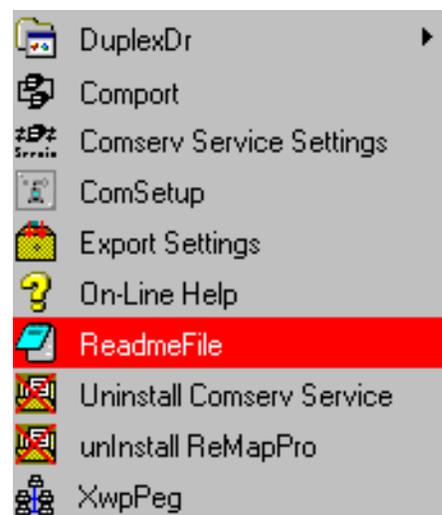
- **Setup complete**

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 ReMapPro Programs' Folder

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



The **DuplexDr** item opens a submenu of four tools that you can use to install, uninstall, configure, and trace the DuplexDr package. (Refer to section **Using DuplexDr** in chapter **DuplexDr**.)

The **Comport** item brings up the **COMPORT** window where you can make settings for serial ports and control their activity. (Refer to section **Using Comport** in chapter **Comport**.)

The **Comserv Service Settings** item brings up the **COMSERV** window where you can make settings for serial ports and control their activity. (Refer to section **Using Comserv** in chapter **Comserv**.)

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

The **Export Settings** item is used to store all the settings made on the COM-TCP/IP connection channel tabs of the **COMPORT** and **COMSERV** windows into the **rmapini.dat** file. (Refer to section **Export Settings** in chapter **Comport**.)

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 ReMapPro package.

The **Uninstall Comserv Service** item is used to remove Comserv from the service list on your system. (Refer to section **Uninstalling Comserv** in chapter **Comserv**.)

The **Uninstall ReMapPro** item is used to uninstall the ReMapPro package from your computer.

(Refer to section **Running UNINSTALL** in chapter **Installing ReMapPro**.)

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 ReMapPro**.)

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.

4. Installing ReMapPro

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



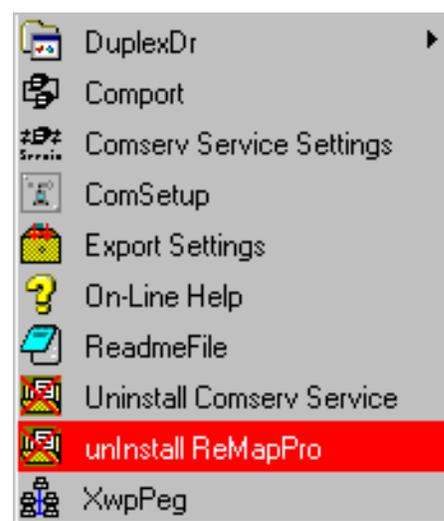
Copyright © 1999 - 2007 Labtam™ Inc.

4. Installing ReMapPro

[< previous](#) | [content](#) | [next >](#)

Running UNINSTALL

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



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

When **Uninstall ReMapPro** 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 **ReMapPro** will be removed from your system.

4. Installing ReMapPro

[< previous](#) | [content](#) | [next >](#)

4. Installing ReMapPro

[< previous](#) | [content](#) | [next >](#)

Upgrading ReMapPro

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.

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.

In this box, 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.

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 ReMapPro detected.

4. Installing ReMapPro

[< previous](#) | [content](#) | [next >](#)

4. Installing ReMapPro

[< previous](#) | [content](#) | [next >](#)

Multi-user Installation

Multi-user installation is intended for installing a single copy of ReMapPro on a disk that will be shared by multiple users. ReMapPro must be configured for each user on each PC it will be used on. Corresponding files that define user's local ReMapPro 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 ReMapPro).

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 ReMapPro 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 ReMapPro created at the first step.

4. Installing ReMapPro

[< previous](#) | [content](#) | [next >](#)

[< previous](#) | [content](#) | [next >](#)

5. Configuring ReMapPro

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

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

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)

5. Configuring ReMapPro

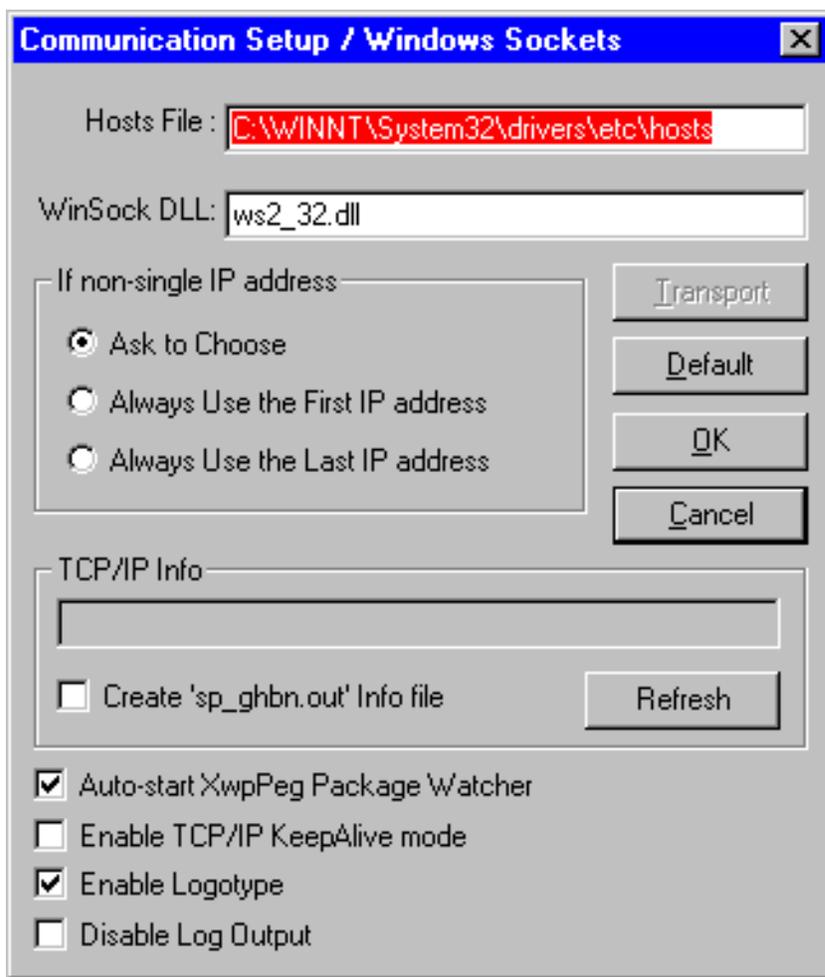
< [previous](#) | [content](#) | [next](#) >

Using ComSetup

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



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



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 ReMapPro'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.

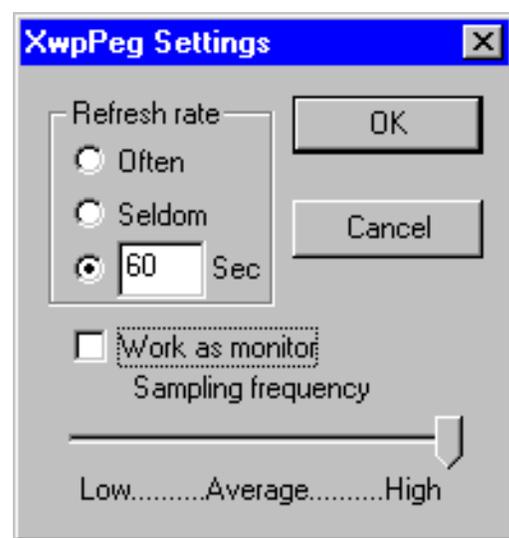
Users can independently launch XwpPeg (before starting any package's application) to check/allocate one license account in advance. If the user starts XwpPeg manually then he/she must also manually terminate it to free the package license account allocated. Since XwpPeg is only informational, users can 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 ReMapPro Programs' folder (i.e., from the Start/Programs/ReMapPro 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:



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).

5. Configuring ReMapPro

< [previous](#) | [content](#) | [next](#) >

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



Copyright © 1999 - 2007 Labtam™ Inc.

5. Configuring ReMapPro

[< previous](#) | [content](#) | [next >](#)

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 ReMapPro.

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 ReMapPro 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 ReMapPro

[< previous](#) | [content](#) | [next >](#)

5. Configuring ReMapPro

[< 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 ReMapPro'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 ReMapPro

[< previous](#) | [content](#) | [next >](#)

6. Comport

This chapter describes how to start and use the Comport program supplied with ReMapPro.

Comport 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, Comport can receive serial data from a COM port of your PC and then send them out to a TCP/IP network through a TCP/IP socket connection on your PC. Comport 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.

Comport makes serial data from your PC available on TCP/IP network and makes TCP/IP data available on serial ports of your PC. This makes it possible for any other computer on the same network to send and receive serial data through a remote serial port on the PC where Comport is running. You can connect to the remote serial port by connecting to a TCP/IP port. Comport can also be used to pass serial data across a corporate intranet or over the Internet.

You could connect a serial device to a COM port on your PC, run Comport 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.

Over TCP/IP-based networks, Comport can function as a TCP/IP server or as a TCP/IP client. While running on your PC in the client mode, Comport establishes a connection with a remote TCP/IP server and then transmits and receives serial data to/from that server. While running on your PC in the server mode, Comport will establish a connection, and then transmit and receive serial data to/from a remote client upon the client's requests.

Comport can be installed on several PCs simultaneously. This allows you to attach a serial device to one PC, have Comport transmit the data across your TCP/IP network to a second PC where another instance of Comport would make the data available at a serial port. You can run Comport as a server on one PC in a network and then run a second copy of it as a client on another PC connecting back to the first copy. Any data that goes in the serial port on the server end will go out the serial port on the client PC and vice versa.

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 Comport for several ports at a time and associate each serial port to a different TCP/IP port either as a TCP/IP client or as a server. Comport can perform the same function as a terminal server except it runs on a PC with a network adapter and one or more serial ports.

You can use Comport as a TCP/IP server with multiple COM ports on a single IP port (so that multiple

modems could be used on the same IP port with auto-switching from one modem to the next).

With Comport, you can create "virtual COM ports" and use "the virtual serial ports pool" in your system. 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. The VSP pool can be used to supply any software application with COM ports (e.g., this is vitally necessary on some modern notebooks that do not have physical COM ports at all). This feature is available after installation of the VSP driver included in the package. With the VSP driver installed, you can configure Comport 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.

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



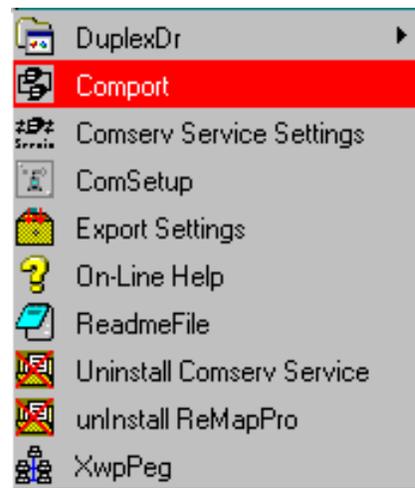
Copyright © 1999 - 2007 Labtam™ Inc.

6. Comport

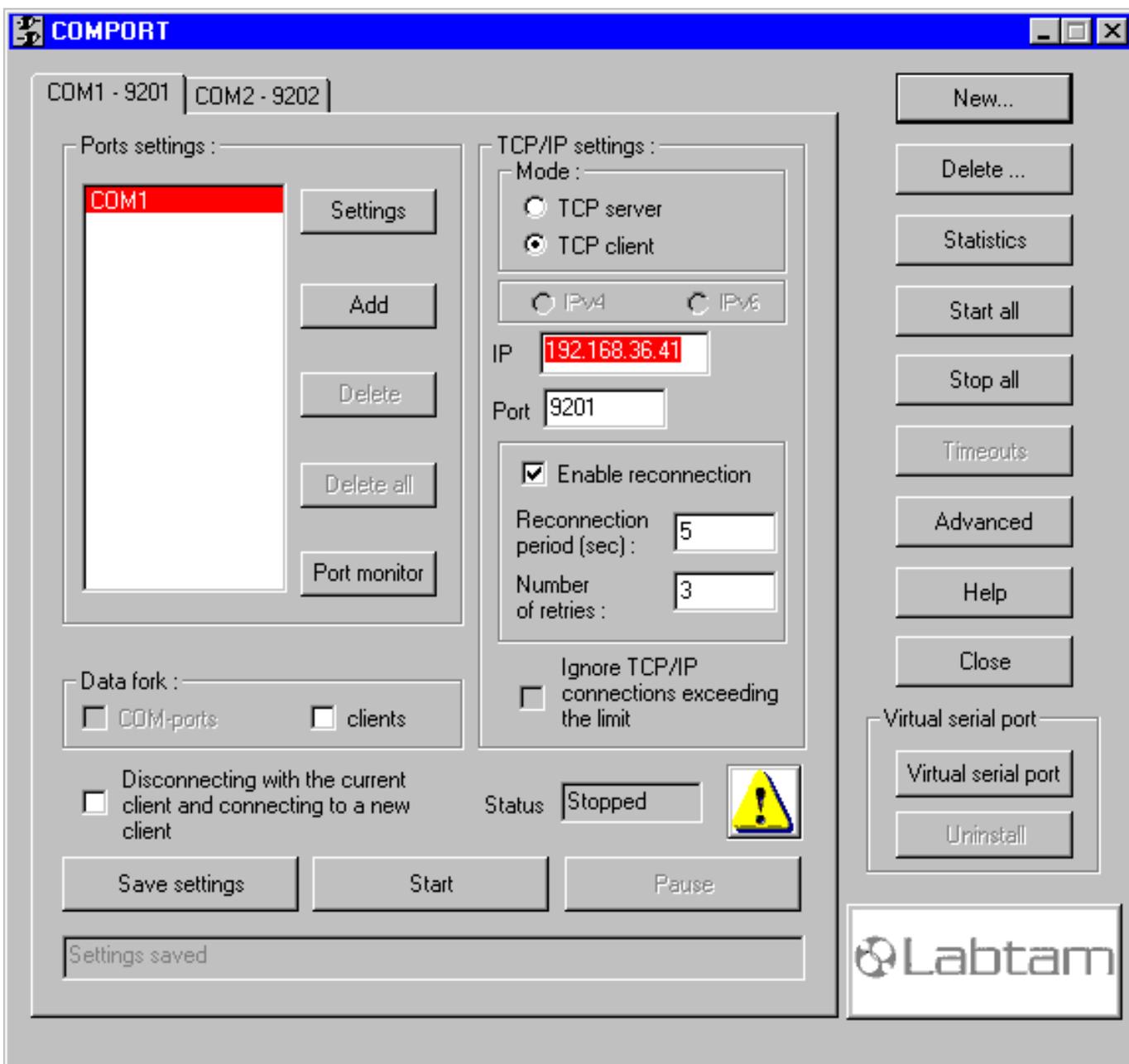
[< previous](#) | [content](#) | [next >](#)

Using Comport

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



The **COMPORT** dialog box will appear on your display:



Before you start a Comport session, you should first have **Port settings** and **TCP/IP settings** defined in a COM-Port tab. The COM-Port tab's header shows the COMx-Port pair (one serial-network data connection channel). One or more COM-Port tabs make up the COM-Port table. You can configure Comport for several serial ports at a time and associate each serial port with a different TCP/IP port either as a TCP/IP client or as a TCP/IP server. On how you can specify **Port settings** and **TCP/IP settings**, refer to section **Configuring Comport**.

To manage the COM-Port table, you can use the **New** and **Delete** buttons. By clicking the **Save settings** button, you can store the current COM-Port tab in the **comport.dat** file located in the home directory of ReMapPro.

You can launch the Comport session for the current COM-Port tab by using the **Start** button, and then you can **Pause** or **Resume** it. To make a COM-Port tab current, just click on its header.

New

This button opens a new COM-Port tab for you to configure a new connection channel configuration.

Delete

Removes the current COM-Port tab (that you see at the top of the window) from the COM-Port table (in the virtual memory).

Start all

This button starts all the connections that are specified in the COM-Port table.

Stop all

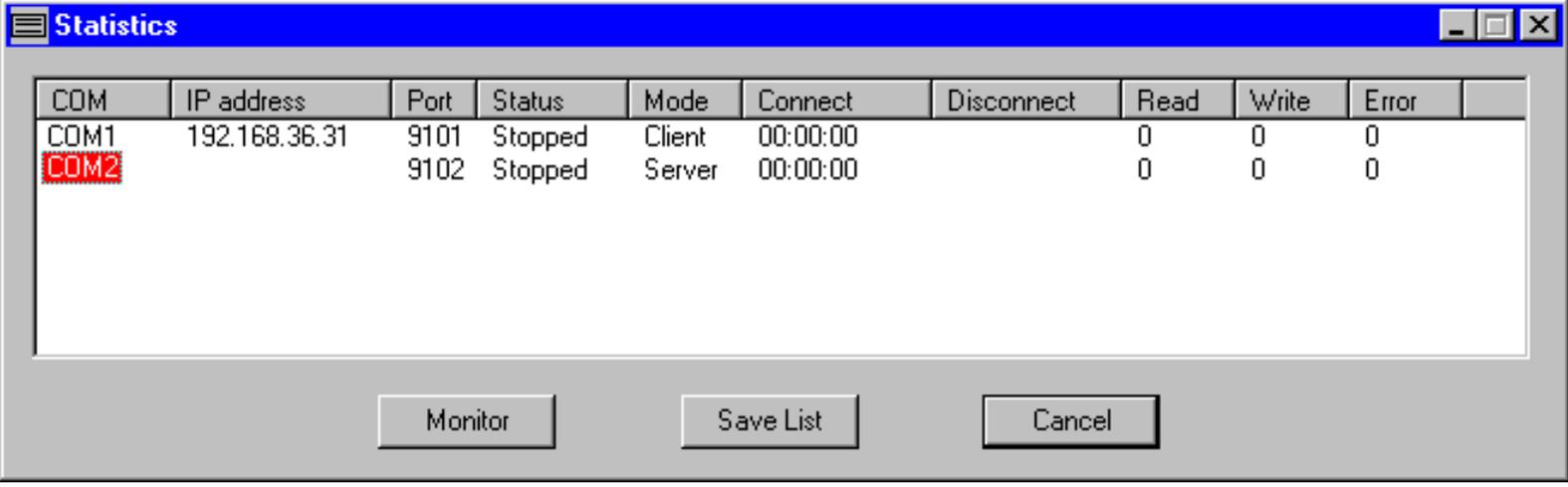
This button stops all the connections that are running (i.e. started).

Close

Stops all the started channels, disconnects all the established connections, and exits Comport.

Statistics

This button is used to display the **Statistics** window to show the status information for each configured COM-Port pair (i.e. for all running sessions):



The screenshot shows a window titled "Statistics" with a table of connection data. The table has columns for COM, IP address, Port, Status, Mode, Connect, Disconnect, Read, Write, and Error. Two rows are visible: COM1 (Client, 192.168.36.31, 9101) and COM2 (Server, 192.168.36.31, 9102). Both are "Stopped". Below the table are three buttons: "Monitor", "Save List", and "Cancel".

COM	IP address	Port	Status	Mode	Connect	Disconnect	Read	Write	Error
COM1	192.168.36.31	9101	Stopped	Client	00:00:00		0	0	0
COM2	192.168.36.31	9102	Stopped	Server	00:00:00		0	0	0

You can select a COM port and press the **Monitor** button to display the **COM-port monitor** window for the port and then watch its activity. (Refer to section **Monitoring COM Ports** below.)

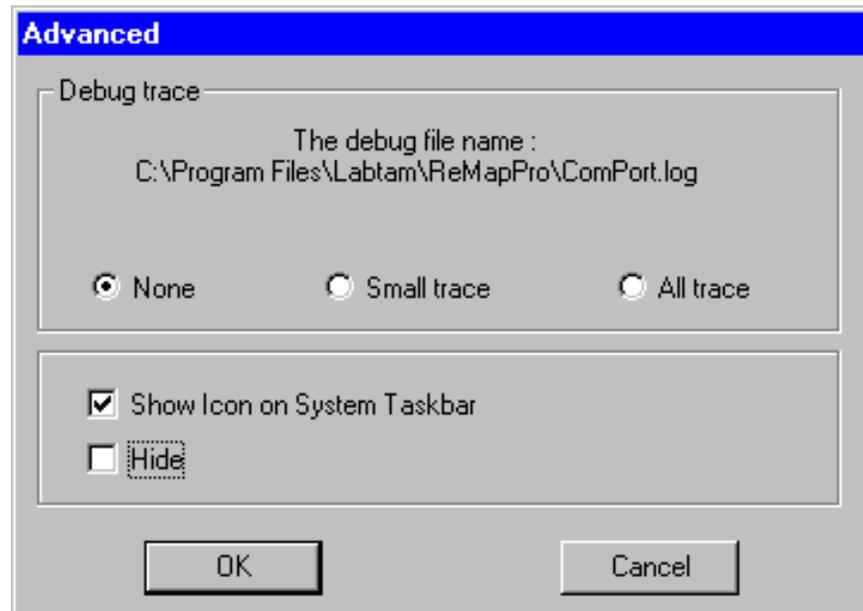
The utility allows the user to display and print a list of all physical ports, redirected ports, and un-configured ports with their corresponding IP address and TCP port number plus an abbreviated description and port status.

You can press the **Save List** button to store the connection channel lines displayed in an ASCII file and then print it.

By clicking Column Headers you can sort information as desired.

The Advanced Box

Clicking the **Advanced** button will come up with the **Advanced** dialog box that lets you specify your preferences for all running connection channels:



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

Show Icon on System Taskbar

Specifies to show the Comport icon on the system taskbar. With this icon, you can exit Comport, configure Comport (i.e. to display the **COMPORT** window), or display the **Statistics** window.

Hide

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

The Debug Trace Options Box

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

None

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

Small trace

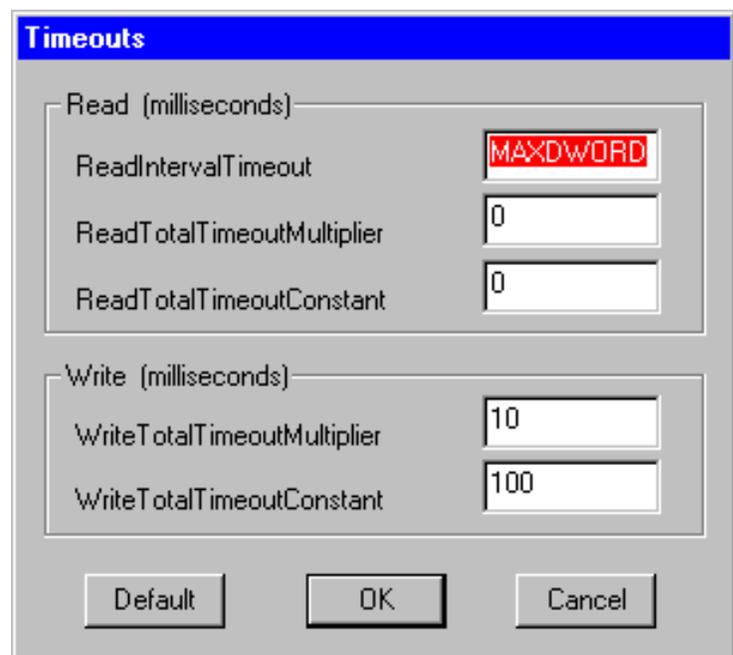
Specifies to store brief debug information in the debug file shown.

All trace

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

Timeouts

This button is used to display the **Timeouts** dialog box to show or set a number of timeout parameters for virtual serial ports:



When you press **Start** the values you specify in this dialog box will begin to take effect for all serial channels (and its ports) that you start.

Note that normally the **Timeouts** button appears dimmed. The **SetTimeouts** setting in the ini-file toggles its state.

The **Default** button is used to reset the timeout parameters to their default values.

By clicking **OK**, any new settings you make to the **Timeouts** dialog box will be saved and the dialog will be closed.

Cancel quits the dialog with no change.

According to the MSDN resources, the timeout parameters are used to set and query the time-out parameters for a communication device. The parameters determine the behavior of the software function operations (i.e. ReadFile and WriteFile) on the device.

ReadIntervalTimeout

Specifies the maximum acceptable time, in milliseconds, to elapse between the arrivals of two characters on the communication line. During a ReadFile operation, the time period begins when the first character is received. If the interval between the arrivals of any two characters exceeds this amount, the ReadFile operation is completed and any buffered data is returned. A value of zero indicates that interval time-outs are not used.

ReadTotalTimeoutMultiplier

Specifies the multiplier, in milliseconds, used to calculate the total time-out period for read operations. For each read operation, this value is multiplied by the requested number of bytes to be read.

ReadTotalTimeoutConstant

Specifies the constant, in milliseconds, used to calculate the total time-out period for read operations. For each read operation, this value is added to the product of the ReadTotalTimeoutMultiplier parameter and the requested number of bytes.

A value of zero for both the ReadTotalTimeoutMultiplier and ReadTotalTimeoutConstant parameters indicates that total time-outs are not used for read operations.

WriteTotalTimeoutMultiplier

Specifies the multiplier, in milliseconds, used to calculate the total time-out period for write operations. For each write operation, this value is multiplied by the number of bytes to be written.

WriteTotalTimeoutConstant

Specifies the constant, in milliseconds, used to calculate the total time-out period for write operations. For each write operation, this value is added to the product of the WriteTotalTimeoutMultiplier parameter and the number of bytes to be written.

A value of zero for both the WriteTotalTimeoutMultiplier and WriteTotalTimeoutConstant parameters indicates that total time-outs are not used for write operations.

Remarks

If an application sets **ReadIntervalTimeout** and **ReadTotalTimeoutMultiplier** to MAXDWORD and sets **ReadTotalTimeoutConstant** to a value greater than zero and less than MAXDWORD, one of the following occurs when the ReadFile function is called:

- 1) If there are any characters in the input buffer, ReadFile returns immediately with the characters in the buffer.
- 2) If there are no characters in the input buffer, ReadFile waits until a character arrives and then returns immediately.
- 3) If no characters arrive within the time specified by **ReadTotalTimeoutConstant**, ReadFile times out.

You should enter the value of **-1** as MAXDWORD in the timeout entry fields.

Running a Connection Channel

When you have a COM Port tab configured (i.e. the Ports settings and TCP/IP settings have correctly been made) you can start/stop/pause/resume the connection channel and watch its status and activity using **Port**

monitor, **Statistics**, or **Advanced** buttons.

Status

This field is used to display the status (stopped/started/paused/resumed) of the current connection channel (i.e. for the current COM Port tab you see at the top of the Comport window).

Save settings

This button is used to store all settings you made in the current COM Port tab in the **comport.dat** file located in the installation directory.

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

Start

Launches the Comport's session with the settings from the current COM-Port tab.

After you have specified the parameters that you would like to use, you can press **Start** to establish a connection between the serial port and the TCP/IP port. The button changes to **Stop** and vice versa.

If the current tab is set up as a TCP/IP client, Comport will immediately try to establish a connection to the specified remote server. If the server is not available, Comport will continually try to establish the connection until it is successful.

If the tab is set up as a TCP/IP server, Comport will wait in "Listen Mode" until a client establishes a connection to it.

Pause

Suspends Comport for the current COM-Port tab. The button changes to **Resume** and vice versa.

The Data fork Box

You can specify how Comport 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 Comport and if you have only one serial port seen in the **Port settings list**.

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

If the **Clients** check box is selected, then Comport will in turn redirect the incoming data flow (i.e., received from each network client connected to Comport) to the serial port (intermixing as

information goes). Also, when the serial port sends data to the network, Comport will in turn duplicate the outgoing data flow to each network client connected to Comport.

- **COM-ports**

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

If the **COM-ports** check box is off, then Comport allows only one network client (one that first connected to Comport) to interact with all the redirected serial ports from the list. Other clients (connected to Comport) will wait until freeing a port. Comport 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, Comport 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 Comport will give all the serial ports from the list to all clients connected to Comport (one free port to only one client).

Disconnecting with the current client and connecting to a new client

This option is available for the server mode of Comport and with a single serial port redirected to a TCP/IP port.

If this check box is enabled, then the Comport's server will connect to a new client (if it detects one) disconnecting with the currently connected client.

6. Comport

< [previous](#) | [content](#) | [next](#) >

Virtual Serial Ports

With Comport, you can create "virtual COM ports" and use "the virtual serial ports pool" in your system. 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. The VSP pool can be used to supply any software application with COM ports (e.g., this is vitally necessary on some modern notebooks that do not have physical COM ports at all). This feature is available after installation of the corresponding VSP driver included in the package. With the VSP driver installed, you can configure Comport 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.

Installing the Virtual Serial Port Driver

When you press the **Virtual serial port** button in the **Comport** window (with the VSP driver not installed), the following dialog box brings up:



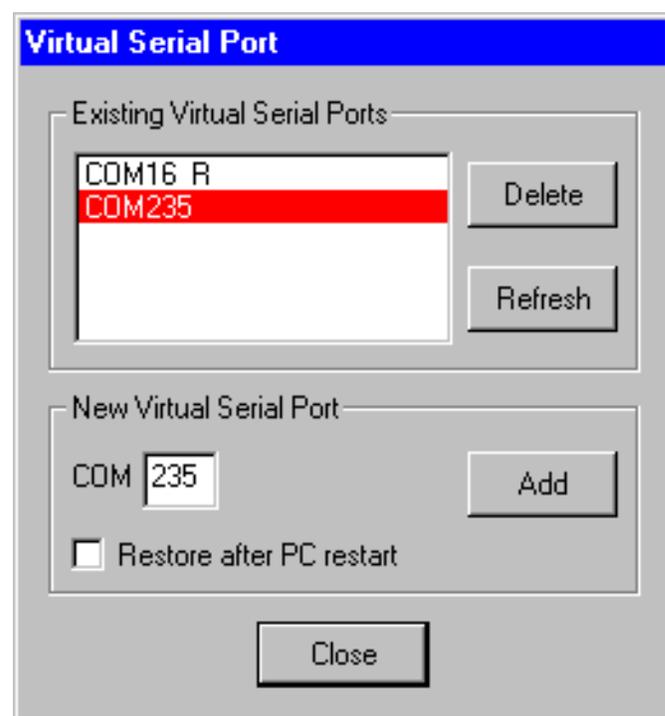
Click **Yes** to install the VSP driver. After successful installing the driver, the following dialog box brings up:



Click **OK** to continue. To activate the VSP driver, you must restart your PC.

Configuring the Virtual Serial Port Driver

When you press the **Virtual serial port** button in the **Comport** window (with the VSP driver installed), the following dialog box brings up:



You can create a new virtual serial port in the pool by entering a port number in the **COM** edit field and pressing the **Add** button in the **New virtual serial port** box.

With the **Restore after PC restart** check box enabled, the added port is marked with the **R** suffix. This means that the port will be stored and loaded next session (so you will see it in the list).

To remove a virtual serial port from the list, highlight it and press the **Delete** button in the **Existing virtual serial ports** box.

The **Refresh** button allows you to restore the virtual serial port list to its initial (when beginning the session) state (after you have made some Add/Delete operations on the list).

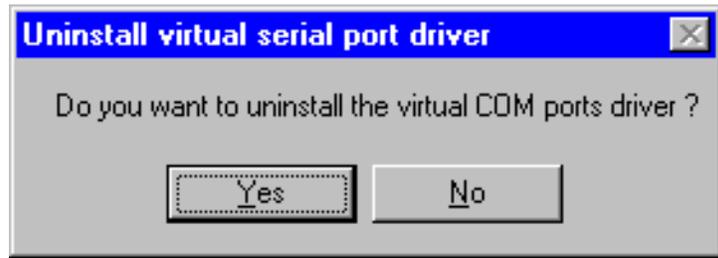
Close saves your settings and applies them immediately.

After you configured the VSP driver, you can launch your software applications to use the settings. For example, you could configure Comport as "connected" to virtual COM6 and configure the VSP driver with the COM6 port in the **Existing Virtual Serial Ports** box. This would get virtual port COM6 "connected" to your TCP/IP network.

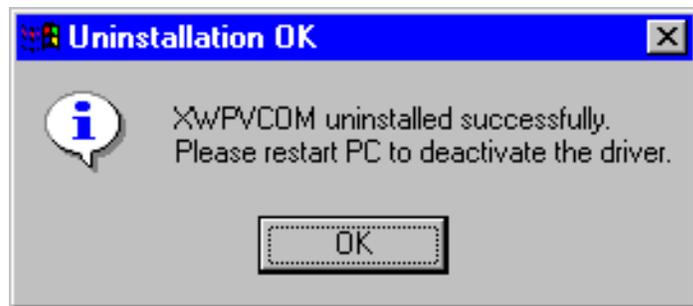
The standard version of the VSP driver allows you to enter up to 10 virtual serial ports and the advanced one – up to 255 (according to your order).

Uninstalling the Virtual Serial Port Driver

When you press the **Uninstall** button in the **Virtual serial port** box of the **Comport** window (with the VSP driver installed), the following dialog box appears:



Click **Yes** to remove your pool of virtual serial ports from the memory and uninstall the VSP driver as well. After successful removing the driver, the following dialog box brings up:



Click **OK** to continue. To deactivate the VSP driver, you must restart your PC.

6. Comport

[< previous](#) | [content](#) | [next >](#)

Configuring Comport

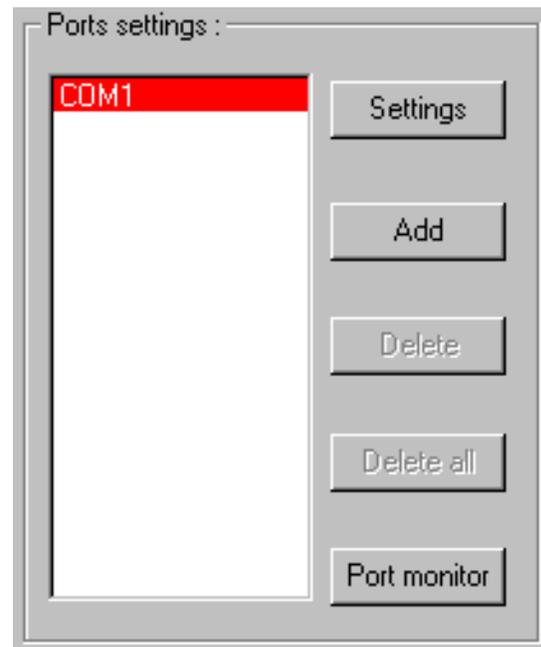
To configure Comport for a connection channel, you should specify **Port settings** and **TCP/IP settings** within a COM-Port tab. You can define several COM-Port tabs to form the COM-Port table.

Serial Port Settings

In this section, you set the parameters for a serial port (COM1, COM2, etc) on your system for Comport to use (i.e. the connection preferences that determine how data is sent and reassembled by your serial devices).

The **COM-port** tab has the **Ports settings** box where you can enter or select a number of serial ports you want to configure (with an attached device). Also, there are three buttons (**Add**, **Delete** and **Delete all**) you can use to create COM port array.

Available serial ports you can add are either one of physical COM ports on your PC or one of virtual serial ports known to your system (from the pool you created with the VSP driver or other tools).



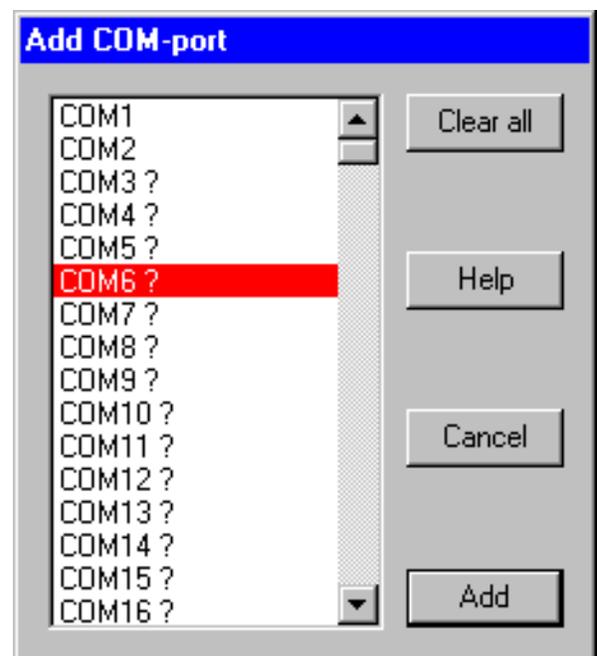
Settings

This button displays the **COMx settings** window for the serial port selected from the current COM

port array displayed (COMx). It is described below.

Add

This button displays the **Add COM-port** dialog box that lets you select serial ports and add them to the array of COM ports.



The ? sign marks serial ports that are absent in your system.

The # sign marks serial ports that are currently in use.

The word **Virtual** marks virtual serial ports in your system.

Clear all clears all selections in the list.

Add closes the box and copies ports selected to the array of COM ports.

Note that you may only copy (and have in the COM ports array) one selected port for the TCP/IP client mode.

Cancel closes the box with no changes.

Delete

This button removes COM ports selected from the current COM port array displayed.

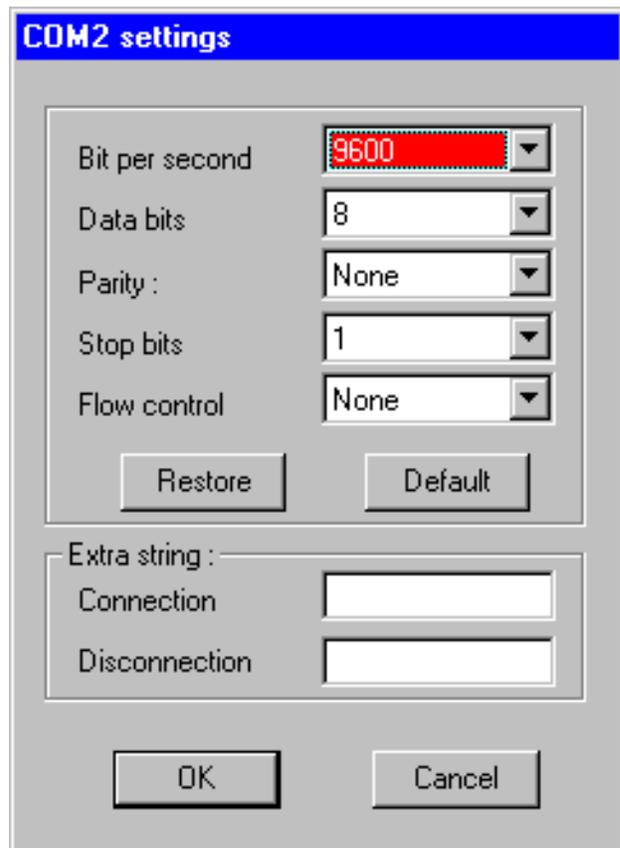
Delete all

This button removes all COM ports from the current COM port array displayed.

Port monitor

This button displays the **COM-port monitor** dialog box that lets you control data logging for a serial port selected. (Refer to section **Monitoring COM Ports** below.)

The COM Port Settings Box



This dialog is used to set up the configuration parameters for the current COM-port to their default or desired values. These are **Bit 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.

Restore

You can use this button to discard values you have just changed and to restore your previous configuration settings (from the **comport.dat** file located in the installation directory) for the current 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**.

Bit per second

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

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 7. 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 **Even**. 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. 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 Comport 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.

TCP/IP Settings

In this section, you set the parameters for a remote TCP connection Comport will use on your system while running a particular serial-to-network session. General TCP/IP settings are described in chapter **Using ComSetup**. For more information, refer to **Appendix A. TCP/IP Connection Overview**.

The following entry fields are available in the **TCP/IP settings** box to configure a network part of the connection channels:

TCP/IP settings :

Mode :

TCP server

TCP client

IPv4 IPv6

IP

Port

Enable reconnection

Reconnection period (sec) :

Number of retries :

Ignore TCP/IP connections exceeding the limit

IP

Specifies an IP address of the remote TCP/IP server on the network Comport will communicate with (as a TCP/IP client only).

Port

This field specifies a TCP/IP port number Comport will use for the remote connection (i.e. a socket number).

If you configure Comport (i.e. a COM port) to act as a TCP/IP client, then you will need to specify the IP address and the port number of the remote TCP/IP server that you want to connect to.

If you configure Comport (i.e. a COM port) to act as a TCP/IP server, then the IP address of the local PC will be used and you only need to specify the local TCP/IP port number that you would like to use.

You can specify any port number between 1 and 65535 which does not cause conflicts between services and Comport. To prevent conflicts between services and Comport, 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.)

Ignore TCP/IP connections exceeding the limit

With this check box enabled, Comport will immediately close those TCP/IP client connections that cannot be established because of lack of available free COM ports in the pool (i.e. Comport will ignore the connection requests instead of enqueueing them). (For the TCP/IP server mode only.)

The max number of available COM ports in the virtual serial ports pool (or the COM port array) may be changed. (See section **The "[COMPORT]" Section of the ini-file** below.)

The Mode Box

For the serial port you specify in the Com port tab, you should also specify the mode in which Comport will communicate with a remote system. Comport can function as either a TCP/IP client or a TCP/IP server.

TCP server

In this mode, Comport will wait for remote clients' requests to a TCP/IP port you specified in the **Port** field (to establish a connection with a remote system, and then to convert and transfer serial data to/from a connected TCP client).

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

Note: As a TCP/IP server, Comport allows to accept only a single client connection to a TCP/IP port. Incoming data received from the client is transmitted out the serial port as it is received. Incoming serial data is sent to a connected TCP/IP client. This enables fully bi-directional communications. Comport allows multiple sessions to run simultaneously but each with different COM-port.

You can use Comport as a TCP/IP server with multiple COM ports (in the COM port array) on a single IP port (so that multiple modems could be used on the same IP port with auto-switching from one modem to the next).

TCP client

In this mode, Comport will establish a connection with a remote system you specified in the **IP** field, and then convert and transfer serial data to/from the remote TCP server through the TCP/IP port you specified in the **Port** field.

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

Note: As a TCP/IP client, Comport allows to establish a connection to a single TCP/IP server. 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. Comport allows multiple sessions to run simultaneously but each with different COM-port.

Note: as a TCP/IP client, Comport initiates the connection(s) to the print server(s) or serial server(s) based on the predefined port mapping. In auto-start mode, the connection is initiated when the mapping program is started. 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 COM port is not available and

ask if the user wants to keep retrying (the message can be disabled if desired).

The Enable Reconnection Box

With this check box enabled, Comport will initiate periodical attempts to reconnect the TCP/IP client connection that gets closed (for the TCP/IP client mode only).

This option is useful when any disruption in connection will cause Comport's TCP/IP client to automatically go to stop. If the option is disabled, the client does not restart until someone clicks the **Start** button on the client.

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

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

6. Comport

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



Copyright © 1999 - 2007 Labtam™ Inc.

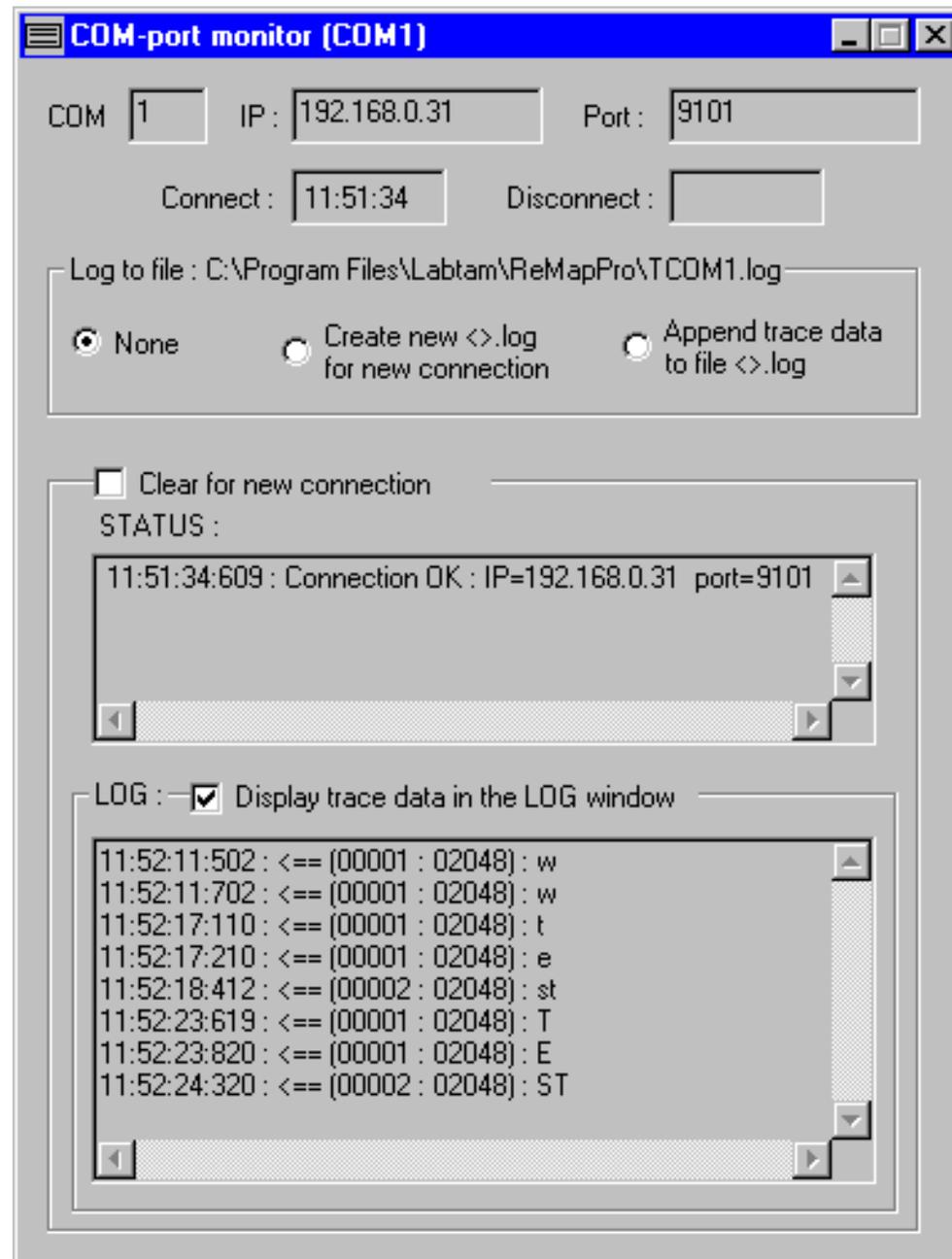
6. Comport

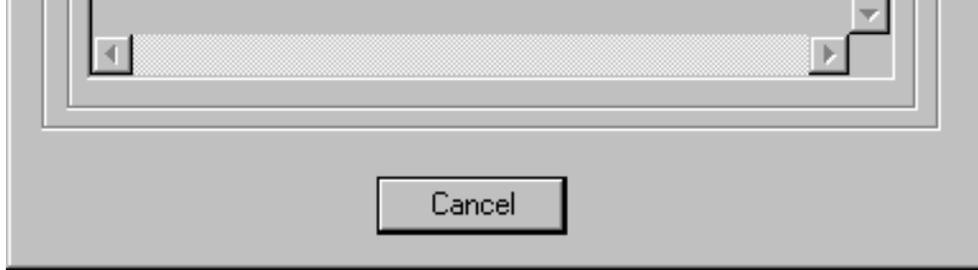
[< previous](#) | [content](#) | [next >](#)

Monitoring COM Ports

The COM-port Monitor Window

From the **Statistics** window, you can select a COM port and press the **Monitor** button to display the **COM-port monitor** window for the port. Then you can watch its activity and control data logging for it. You can display the **COM-port monitor** window also by pressing the **Port monitor** button form the **Port Settings** pane.





COM displays a serial port number for the connection channel.

IP displays an IP address for the connection channel.

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

Connect shows the time when the connection was established (i.e. the channel started).

Disconnect shows the time when the connection was terminated (i.e. the channel stopped).

Cancel closes the **COM-port 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 the new log-file 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.

Append trace data to file <>.log

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

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.

6. Comport

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



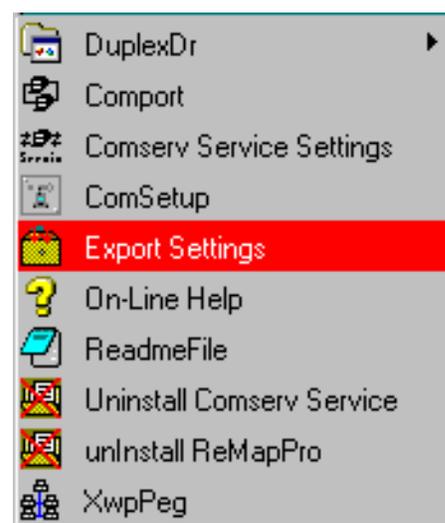
Copyright © 1999 - 2007 Labtam™ Inc.

6. Comport

[< previous](#) | [content](#) | [next >](#)

Exporting Settings

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



you can store all the settings you have made on the COM-TCP/IP connection channel tabs of the Comport and Comserv windows into the **rmapini.dat** file. This ReMapPro configuration file will be created in the home directory and include data from files **comport.dat**, **comserv.dat**, and **xwp.ini** in the ASCII format.

With **Export Settings**, you can create several ReMapPro configuration files with different Comport and Comserv settings you need (using renaming of the **rmapini.dat** file before next exporting).

The ReMapPro configuration file, **rmapini.dat**, is used during secondary installations of the ReMapPro 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 files **comport.dat**, **comserv.dat**, and **xwp.ini** from the **rmapini.dat** file (thus exporting settings from one installation site to another one).

6. Comport

[< previous](#) | [content](#) | [next >](#)

6. Comport

[< previous](#) | [content](#) | [next >](#)

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

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

Number_of_COM-ports=N

This entry line specifies Comport to use **N** COM ports (as the max number of COM ports in the pool (or the array) used by the TCP/IP server) instead of the default (64) value.

RtsControlForNone=L

You can select one of three modes for RTS flow control if **Flow control** in the Serial Setting is set to **None**.

Behavior of Comport for these modes is implemented according to the MSDN Library, i.e., the following flags are set up after opening a COM-port:

- 'fInX', 'fOutX', 'fOutXDsrFlow', and 'fOutXCtsFlow' are all set to FALSE
- 'fDtrControl' is set to DTR_CONTROL_ENABLE
- 'fRtsControl' is set to RTS_CONTROL_ENABLE.

To select the mode for the RTS flow control (if **Flow control** in the Serial Setting is set to **None**), the **RtsControlForNone=...** line can be inserted into the "[COMPORT]" section of the **xwp.ini** file (that is in the home directory of the package).

The setting provides the following modes:

- **RtsControlForNone=0**
With this setting, flag 'fRtsControl' is set to RTS_CONTROL_DISABLE after opening a COM-port. RTS_CONTROL_DISABLE disables the RTS line when the device is opened and leaves it disabled.
- **RtsControlForNone=1**
With this setting or if this line is absent, the software works as in the previous version, i.e. flag 'fRtsControl' is set to RTS_CONTROL_ENABLE after opening a COM-port. RTS_CONTROL_ENABLE enables the RTS line when the device is opened and leaves it on.
- **RtsControlForNone=2**
This setting makes 'GetCommState' before 'SetCommState' without changing parameter

'PortDCB.fRtsControl' for RTS flow control.

To take the mode into effect, you should do the following:

1. Stop port mapping
2. Select the mode of your choice
3. Start port mapping again.

SetTimeouts=L

With L=1, this entry line causes Comport to show the **Timeouts** button active in the main window (thus allowing you to specify the Timeout parameters).

With L=0, Comport will keep the button dimmed.

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 "ASYN" 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").

Tracing=L

With L=0, Comport will store no tracing data.

With L=1, this entry line specifies Comport to store brief tracing information into the **Comport.log** file.

With L=3, Comport will store the contents of every packet too in the log-file (thus decreasing the performance).

6. Comport

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



Copyright © 1999 - 2007 Labtam™ Inc.

7. Comserv

This chapter describes how to start and use the Comserv program supplied with ReMapPro.

Comserv is a software program that designed to be used as a standard MS Windows NT service. Comserv's functionality and settings are just the same as of Comport. Additionally, you can control the behavior of Comserv as the NT service.

To manage the Comserv's session (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 >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



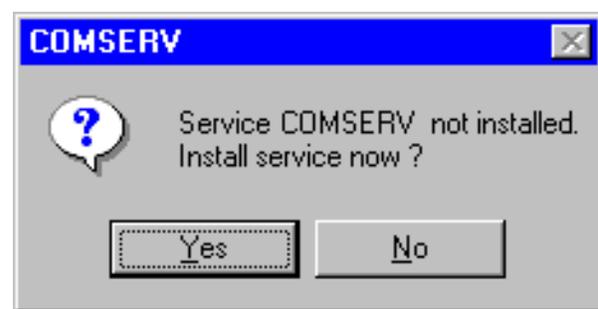
Copyright © 1999 - 2007 Labtam™ Inc.

7. Comserv

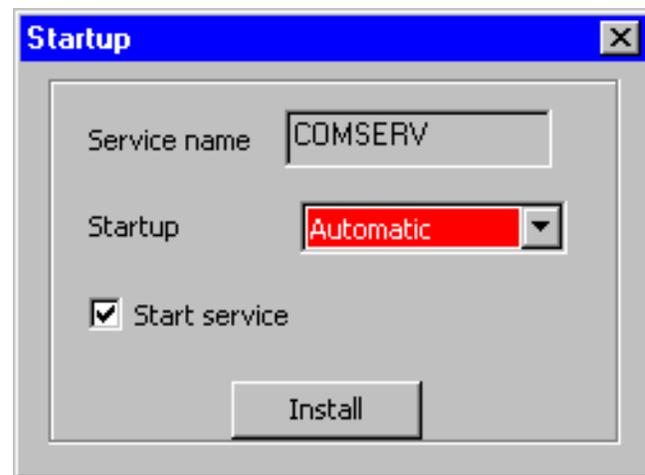
[< previous](#) | [content](#) | [next >](#)

Installing Comserv

If Comserv is not installed when you are trying to use it, then the following box brings up:



Click **Yes** to install Comserv and include it to the service list on your system. Then the **Startup** box will appear:



If you have a typical installation, many services are configured as **Automatic** (that is, they start automatically when the system starts or when the service is called for the first time). If a service is configured as **Manual**, you must start the service manually before it can be loaded by the operating system and made available for use. If a service is configured as **Disabled**, it cannot be started automatically or manually.

The service name is **COMSERV**.

In the **Startup type** list box of the **Startup** box, you should choose how the COMSERV service is started.

Select the **Start service** check box, if you want the service (configured as **Automatic**) to start immediately after its installation, and click **Install**.

You should test that you have configured Comserv to operate successfully in the Normal mode before attempting to run it as a MS Windows Service. (Refer to section **Using Comport**.)

7. Comserv

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



Copyright © 1999 - 2007 Labtam™ Inc.

7. Comserv

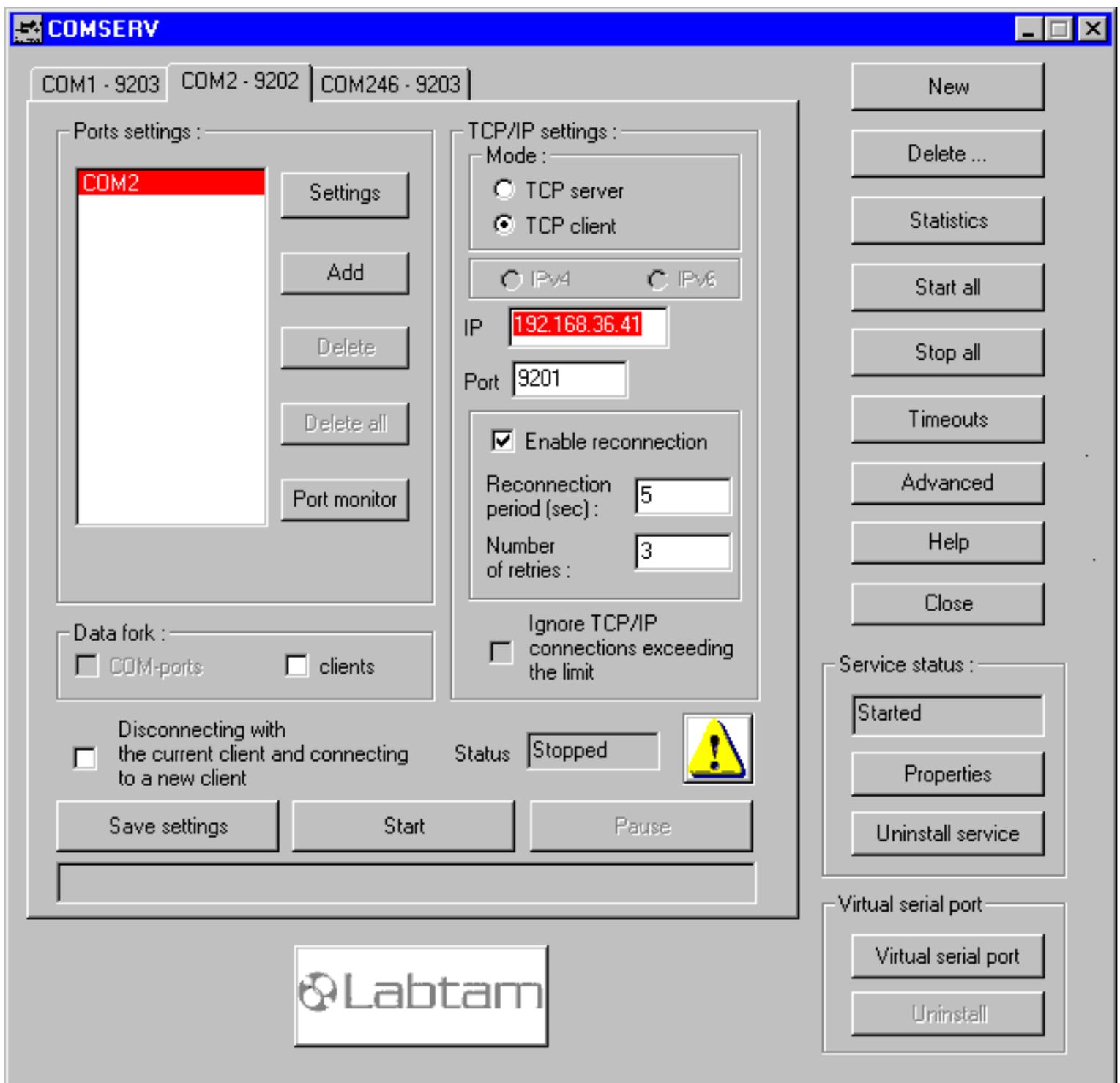
< [previous](#) | [content](#) | [next](#) >

Configuring Comserv

To make settings for Comserv, you can click on the **Comserv Service Settings** item in the ReMapPro Programs' folder (i.e., from the Start/Programs/ReMapPro menu):



With Comserv installed, the **COMSERV** window will appear on your display:



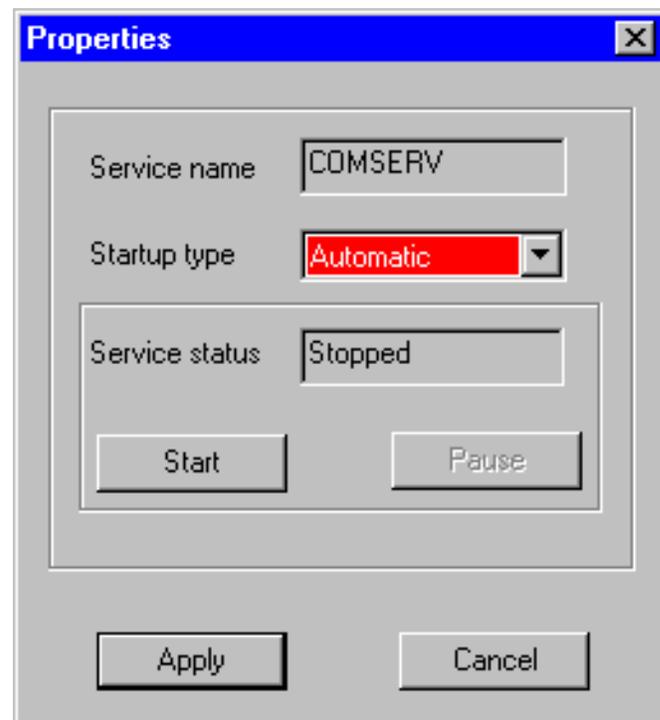
To configure Comserv, you should specify settings just in the same manner as you configure Comport. Additionally, you should control Comserv as a MS Windows service with the **Service status** box.

Note that with the **Save settings** button, you can store all settings you have in the COMSERV COM-Port tabs in the **comserv.dat** file located under the home directory of ReMapPro.

Also, there are some additional settings you can specify to customize Comserv. They are described in section **The "[COMSERV]" Section of the ini-file**.

The Service Status Box

When you press the **Properties** button from within the Service status box on the COMSERV window, the **Properties** dialog box brings up:



If you have a typical installation, many services are configured as **Automatic** (that is, they start automatically when the system starts or when the service is called for the first time). If a service is configured as **Manual**, you must start the service manually before it can be loaded by the operating system and made available for use. If a service is configured as **Disabled**, it cannot be started automatically or manually.

To configure how the COMSERV service is started, you choose the **Startup type** and press the **Apply** button. For example, **Automatic** will start up the service every time you restart your system. If you choose **Manual**, then you can use Control Panel/Administrative Tools/Services of MS Windows to manage the Comserv's session (e.g., to start/stop services as administrator).

You can also manage the service session by pressing the **Start/Stop** and **Pause/Resume** buttons (with the COMSERV service installed), and watching the **Service status** field.

To start, stop, pause, resume, or restart a service, you can also open the **Services** window (clicking **Start/Settings/Control Panel/Administrative Tools/Services**), right-click your service, and then click **Start**, **Stop**, **Pause**, **Resume**, or **Restart**.

Pressing **Cancel** closes the dialog box (with no changes to the service properties).

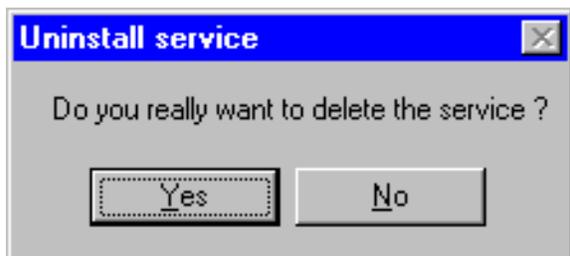
Uninstalling Comserv

You can remove Comserv from the service list on your system by pressing **Uninstall service** from within the Service status box on the COMSERV window.

Also, you can completely uninstall Comserv from your system by choosing the **Uninstall Comserv Service** item from the Start/Programs/ReMapPro menu:



The following dialog box appears:



Click **Yes** to remove Comserv from the service list on your system and then restart your PC.

Note that you must stop all started connection channels (in the **COMSERV** window) before uninstalling Comserv.

Example of running COMSERV as a MS Windows 2000/XP service

Login to MS Windows XP as Administrator and install the ReMapPro package and license file as per instructions.

You should test that you have configured Comserv to operate successfully in the Normal mode before attempting to run it as a MS Windows Service. (Refer to section [Using Comport.](#))

- **Creating the COMSERV Service**

Install Comserv (as described in section [Installing Comserv.](#))

- **Defining the COMSERV's Properties**

1. Open the **Services** window.

To open **Services**, click **Start**, point to **Settings**, and then click **Control Panel**. Double-click **Administrative Tools**, and then double-click **Services**.

2. In the **Services** window, scroll down to **COMSERV**, right-click on the service, and then click **Properties**.
3. On the **General** tab in the COMSERV Properties window, set **Startup type** to **Automatic**.
4. On the **Log On** tab, set the account field to **.\Administrator**.

We prefer to choose the default option, i.e. the **Local System account** radio button, 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.

5. On the **Recovery** tab, click the actions you want to take place when a service fails in **First attempt**, **Second attempt**, and **Subsequent attempts** (e.g., set **First failure** and **Second failure** to **Restart the Service**).

If you select **Run a File**, do not specify programs or scripts that require user input. If you select **Reboot the Computer**, you can specify how long to wait before restarting the computer by clicking **Reboot Computer Information**. You can also create a message to display to remote users before the computer restarts.

We prefer to choose the default option, **Take no Action**, for all the failures (First, Second, and Subsequent).

6. Leave the **Dependencies** tab unchanged.

The top list on the **Dependencies** tab identifies the other services that must be running for the selected service to function.

The bottom list on the **Dependencies** tab identifies the services that require the selected service to be running so they can function.

7. Click **OK** to save your settings and close the COMSERV Properties window.

- **Making the MS Windows Firewall settings**

You must resolve access to Comserv for MS Windows Firewall (this is particularly actual for MS Windows XP Service Pack 2).

1. Run MS Windows Firewall by clicking **Start/Control Panel/Network and Internet Connections/Windows Firewall**
2. Open the **Exceptions** tab
3. Choose **Add program...**
4. Enter a path name for the COM Port Redirector Service executable file (PATH indicates your ReMapPro home directory):

"PATH\comserv.exe"

5. Click **OK**.

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

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

Number_of_COM-ports=N

This entry line specifies Comserv to use **N** COM ports (as the max number of COM ports in the pool (or the array) used by the TCP/IP server) instead of the default (64) value.

RtsControlForNone=L

You can select one of three modes for RTS flow control if **Flow control** in the Serial Setting is set to **None**.

Behavior of Comserv for these modes is implemented according to the MSDN Library, i.e., the following flags are set up after opening a COM-port:

- 'fInX', 'fOutX', 'fOutXDsrFlow', and 'fOutXCtsFlow' are all set to FALSE
- 'fDtrControl' is set to DTR_CONTROL_ENABLE
- 'fRtsControl' is set to RTS_CONTROL_ENABLE.

To select the mode for the RTS flow control (if **Flow control** in the Serial Setting is set to **None**), the **RtsControlForNone=...** line can be inserted into the "[COMSERV]" section of the **xwp.ini** file (that is in the home directory of the package).

The setting provides the following modes:

- **RtsControlForNone=0**
With this setting, flag 'fRtsControl' is set to RTS_CONTROL_DISABLE after opening a COM-port. RTS_CONTROL_DISABLE disables the RTS line when the device is opened and leaves it disabled.
- **RtsControlForNone=1**
With this setting or if this line is absent, the software works as in the previous version, i.e. flag 'fRtsControl' is set to RTS_CONTROL_ENABLE after opening a COM-port. RTS_CONTROL_ENABLE enables the RTS line when the device is opened and leaves it on.
- **RtsControlForNone=2**
This setting makes 'GetCommState' before 'SetCommState' without changing parameter

'PortDCB.fRtsControl' for RTS flow control.

To take the mode into effect, you should do the following:

1. Stop port mapping
2. Select the mode of your choice
3. Start port mapping again.

SetTimeouts=L

With L=1, this entry line causes Comserv to show the **Timeouts** button active in the main window (thus allowing you to specify the Timeout parameters).

With L=0, Comserv will keep the button dimmed.

Tracing=L

With L=0, Comserv will store no tracing data.

With L=1, this entry line specifies Comserv to store brief tracing information into the **Comserv.log** file.

With L=3, Comserv will store the contents of every packet too in the log-file (thus decreasing the performance).

7. Comserv

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



Copyright © 1999 - 2007 Labtam™ Inc.

This page is a part of ReMapPro online Help Manual.

ReMapPro makes serial (RS232) data from your PC available on TCP/IP-based networks and makes TCP/IP data available on COM port on your PC. Visit [Home of ReMapPro](#) for more information.

[< previous](#) | [content](#) | [next >](#)

8. DuplexDr

This chapter describes how to install, uninstall, configure, and use DuplexDr supplied with ReMapPro.

You can use this tool under MS Windows NT4/2K/2K3/XP/XP-x64.

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



Copyright © 1999 - 2007 Labtam™ Inc.

8. DuplexDr

[< previous](#) | [content](#) | [next >](#)

What is in DuplexDr

The DuplexDr package is an integrated software tool consisting of the following functional parts:

- XVCINST.EXE – installs the DuplexDr package
- XVCUNINS.EXE – uninstalls the DuplexDr package
- XVCACTIV.EXE – checks the status of the DuplexDr package
- XVCCONF.EXE – configures the DuplexDr package
- XVCTRC.EXE – sets up trace flags for the DuplexDr package
- XVCDW2K.W2K – the DuplexDr driver for MS Windows 2000
- XVCDW40.W40 – the DuplexDr driver for MS Windows NT4

8. DuplexDr

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)

8. DuplexDr

[< previous](#) | [content](#) | [next >](#)

Using DuplexDr

With DuplexDr, you can create a pair of virtual serial ports (VSP pair) "virtually connected to each other" in your system. A virtual serial port 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. Entering VSP pairs, you can create and use "a VSP pairs pool". The pool of VSP pairs can be used to supply any MS Windows application with serial ports as though they were standard serial ports and physically connected in pairs. This feature is available after installation of the corresponding driver included in the package. This allows MS Windows applications to generate or consume data to and from VSP as though the data were being processed by physical hardware.

DuplexDr can be used for sending and receiving data between two separate MS Windows applications running on your PC, thus connecting one application to another by using a VSP pair (for example, Comport and HyperTerminal). Also, you can use the tracing utility, XVCTRC, to document the virtual serial port traffic between your software applications that communicate with such a VSP pair channel (e.g., for application development testing).

After you configured DuplexDr, you can launch your software VSP applications to use the settings. For example, you could configure one HyperTerminal as "connected" to COM6 and another HyperTerminal as "connected" to COM17 respectively, and configure DuplexDr with the COM6<==>COM17 pair in the **Existing Virtual Serial Port Pairs** box. Because the DuplexDr driver considers virtual serial ports in pairs as "physically connected", these settings would get both HyperTerminals "connected" to each other (through DuplexDr), so all you type in the first terminal will appear at the second one.

If you configure Comport as "connected" to COM6 and HyperTerminal as "connected" to COM17 respectively, these settings would get HyperTerminal "connected" to a TCP/IP network (through DuplexDr and Comport), and so you could "chat" with remote PCs on your network. Another example is when you use Comport instead of HyperTerminal (thus connecting two networks or same with the "COM6<==>COM17 connector").

You can transfer files with any data or control commands through such virtual serial flow connection. You can configure the virtual serial ports to the maximum speed value MS Windows allows.

8. DuplexDr

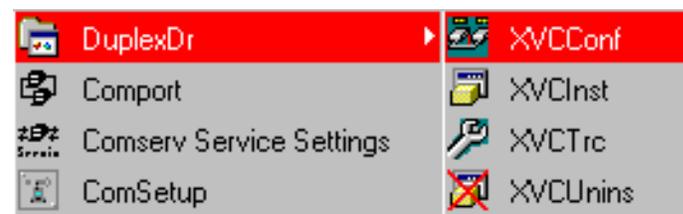
[< previous](#) | [content](#) | [next >](#)

8. DuplexDr

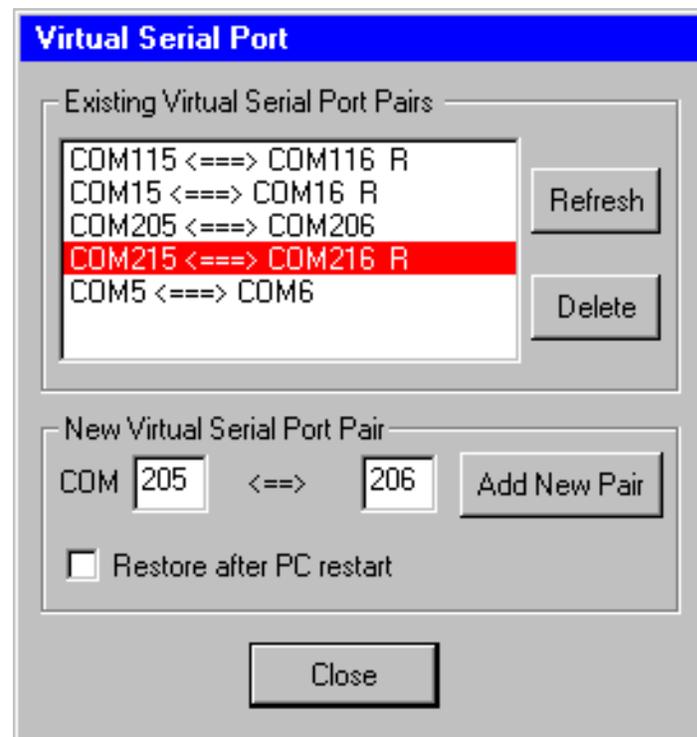
[< previous](#) | [content](#) | [next >](#)

Configuring DuplexDr

To configure the DuplexDr package, you should launch the XVCCONF utility and then specify a number of virtual serial port pairs with or without the Restore property. You can do this by clicking on the **XVCConf** item in the DuplexDr Programs' folder (i.e., from the Start/Programs/ReMapPro/DuplexDr menu):



When you start up XVCCONF (with the DuplexDr driver installed), the following dialog box brings up:



You can create a new virtual serial port pair in the pool by entering port numbers in the **COM** edit fields and pressing the **Add New Pair** button in the **New Port Pair** box. As a result, you will see a new string with serial ports entered. The string means that DuplexDr will create the two virtual serial ports in your system and virtually "connect" them to each other (as though physical ports via null-modem cable), so serial data (in RS-232 protocol) can transfer through such a connection (DuplexDr

will emulate the ports and the process).

With the **Restore after PC restart** check box enabled, the added pair is marked with the **R** suffix. This means that the pairs will be stored and loaded next session (so you will see them in the list).

To remove a virtual serial port pair from the list, highlight it and press the **Delete** button in the **Existing Virtual Serial Port Pairs** box.

The **Refresh** button allows you to restore the virtual serial port list to its initial (when beginning the session) state (after you have made some Add/Delete operations on the list).

Close saves your settings and applies them immediately.

The greatest value for the port number is 254, so the maximal number of VSP pairs you can have in your system is 127.

The standard version of DuplexDr allows you to enter up to 10 virtual serial port pairs and the advanced one – up to 127 (according to your order).

8. DuplexDr

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)



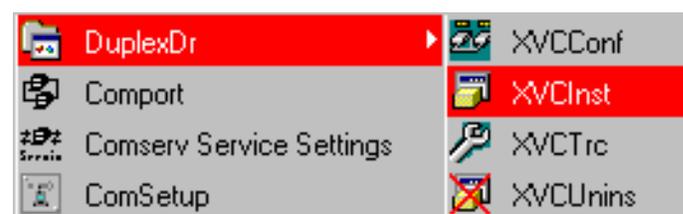
Copyright © 1999 - 2007 Labtam™ Inc.

8. DuplexDr

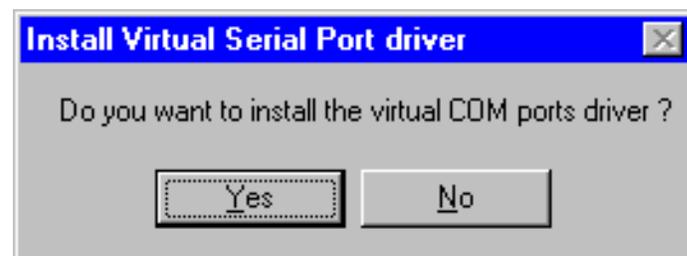
[< previous](#) | [content](#) | [next >](#)

Installing DuplexDr

To install the DuplexDr driver, you should launch the XVCINST utility. You can do this by clicking on the **XVCInst** item in the DuplexDr Programs' folder (i.e., from the Start/Programs/ReMapPro/DuplexDr menu):



When you start up XVCINST (with the DuplexDr driver not installed), the following dialog box will appear:



Click **Yes** to install the DuplexDr driver. After successful installing the DuplexDr driver, the following dialog box brings up:



Click **OK** to continue. To activate the DuplexDr driver, you must restart your PC.

8. DuplexDr

[< previous](#) | [content](#) | [next >](#)

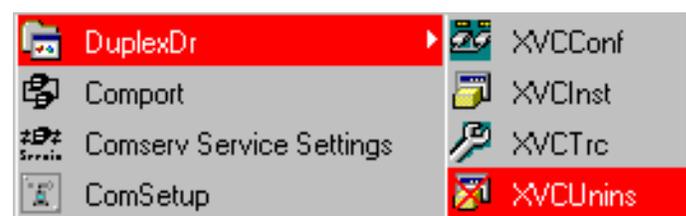


8. DuplexDr

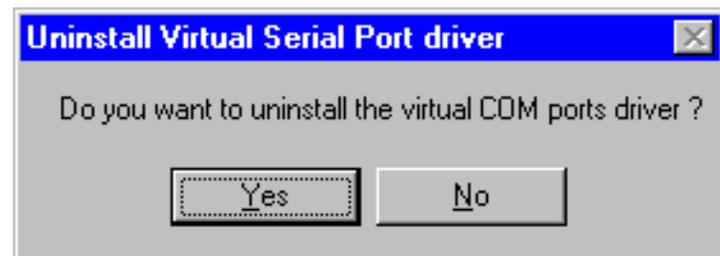
< [previous](#) | [content](#) | [next](#) >

Uninstalling DuplexDr

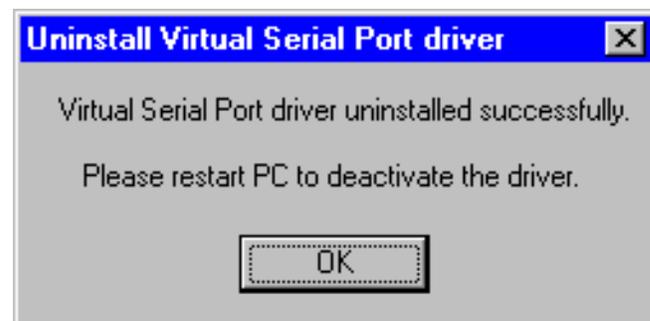
To uninstall the DuplexDr driver, you should launch the XVCUNINS utility. You can do this by clicking on the **XVCUnins** item in the DuplexDr Programs' folder (i.e., from the Start/Programs/ReMapPro/DuplexDr menu):



When you start up XVCUNINS (with the DuplexDr driver installed), the following dialog box appears:



Click **Yes** to remove your pool of virtual serial port pairs from the memory and uninstall the DuplexDr driver as well. After successful removing the DuplexDr driver, the following dialog box brings up:



Click **OK** to continue. To deactivate the DuplexDr driver, you must restart your PC.

8. DuplexDr

< [previous](#) | [content](#) | [next](#) >



8. DuplexDr

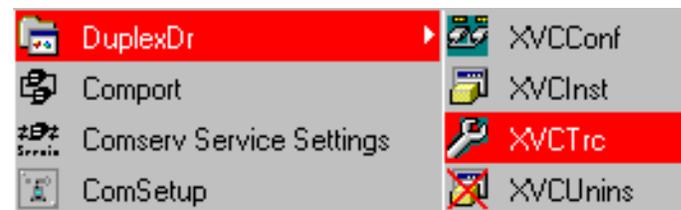
[< previous](#) | [content](#) | [next >](#)

Tracing DuplexDr

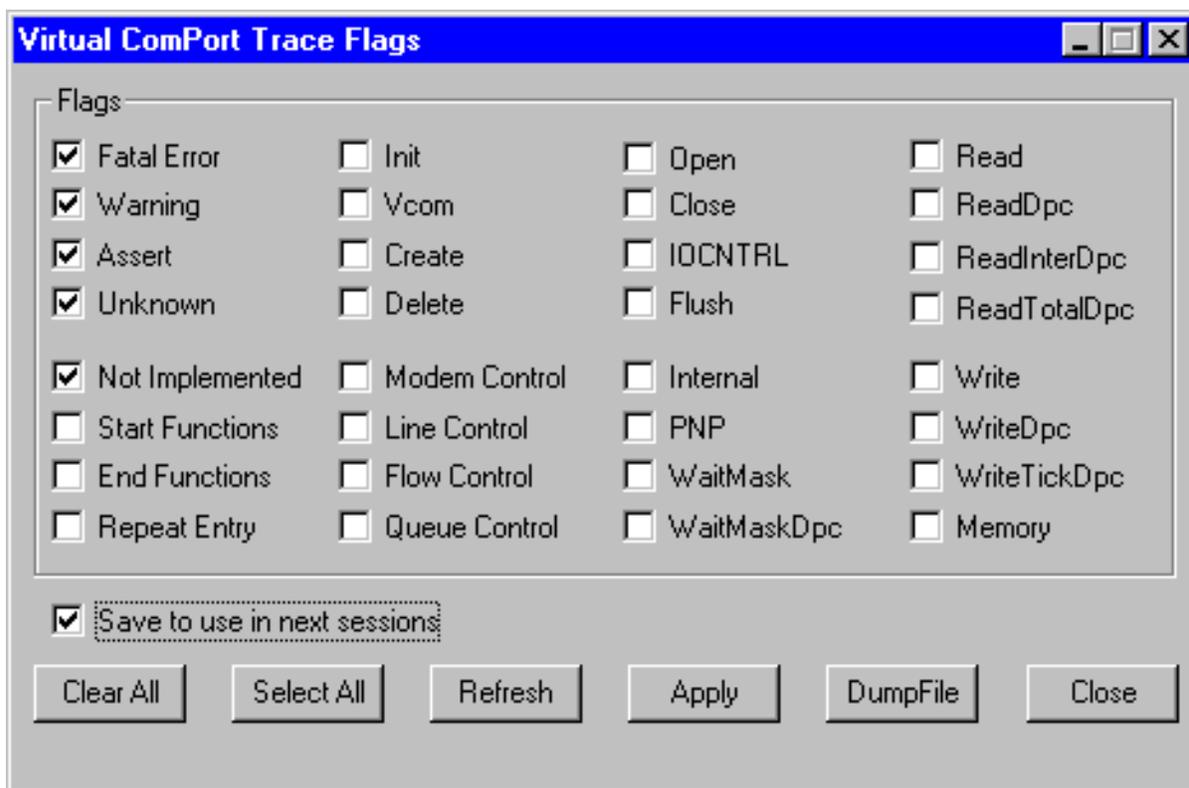
The XVCTRC utility is designed for setting a number of trace flags (by toggling related check boxes to filter out data of your interest) and passing them to the DuplexDr driver. According to the Trace Flags set, utilities of the package will filter out and save related information into corresponding 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 a VSP pair channel.

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 XVCTRC utility is primarily meant for testing and analyzing how your software VSP applications interface to each other while using serial port communication features.

To set up Trace Flags, you should launch the XVCTRC utility. You can do this by clicking on the **XVCTrc** item in the DuplexDr Programs' folder (i.e., from the Start/Programs/ReMapPro/DuplexDr menu):



When you start up XVCTRC (with the DuplexDr driver installed), the **Virtual ComPort Trace Flags** window appears:



If you click the **Close** button, XVCTRC closes the window with no changes.

Pressing the **Apply** button passes the Trace Flags set to the DuplexDr driver and closes the window. The Trace Flags will take effect immediately.

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

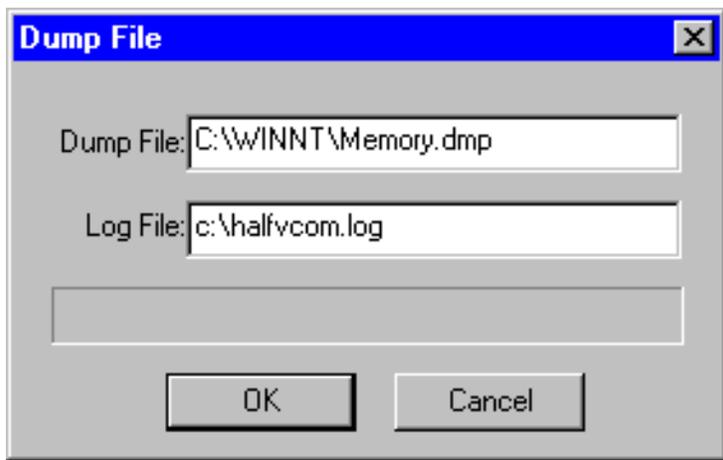
Refresh restores all Trace Flags to their initial states (when loaded the XVCTRC utility or after the latest **Apply**).

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 XVCTRC utility).

Note: with $L=1$, the **Tracing=L** entry line in the ini-file specifies tools to save tracing information into corresponding tracing files. With $L=0$, tools will not write tracing data into the files.

The DumpFile Box

The **DumpFile** button in the **Virtual ComPort Trace Flags** window brings up the dialog box:



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.

8. DuplexDr

[< previous](#) | [content](#) | [next >](#)

[Home](#) | [Product](#) | [Download](#) | [Order Now](#) | [Upgrade](#) | [Support](#) | [Pricing](#) | [Company Information](#) | [Contact Us](#)

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 **ReMapPro** 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 **ReMapPro** may also be either a URL or the name of a computer located on your network. If you are configuring **ReMapPro** 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 **ReMapPro** 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 ReMapPro**.

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](#)

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](#)



Copyright © 1999 - 2007 Labtam™ Inc.