

VirtPort

User Guide

Version 1.4 and later



Rasmussen Software, Inc.

10240 SW Nimbus Ave., Suite L9, Portland, Oregon 97223

(503) 624-0360

www.anzio.com

rsi@anzio.com

Copyright © 2004-2006 by Rasmussen Software, Inc., All Rights Reserved

What is Virtport?

VirtPort is a pair of programs, one for Unix/Linux and one for Windows, that together create a virtual serial port. That is, they make a serial port on the Windows PC appear as a serial port on the Unix/Linux system. This connection happens over a TCP/IP network, using very elementary socket calls.

Once the two programs are set up and running, a Unix/Linux-based program can access the PC's serial port as though it were a Unix/Linux serial port; that is, a `/dev/something` port. Communication can go both ways. However, control operations (ioctl calls) performed by this controlling program to set baud rate, etc., will be ignored; the correct baud rate must be set up on the PC initially.

VirtPort is typically used to drive one or more serial devices attached to a Windows PC acting as a terminal. Examples include scales, pole displays, PIN-pad entry devices, etc. It is NOT intended for printer support; we have more appropriate programs for that. It is also not intended for operating terminals via `getty`.

Requirements

1. On the Unix/Linux side, the person installing virtport must have root privileges. However, once installed, virtport can be run by any user (unless the installed explicitly restricts it using Unix/Linux permission settings).
2. On the Windows side, the program can be installed and run by any user. In fact, there is no formal installation procedure; there is just an EXE file.
3. You will generally want the PC to have a static IP address, although this is not strictly required.
4. The PC's software firewall, if there is one, will need to allow an inbound connection on the appropriate port (default 9100). Any external firewalls will need to allow this also.

Running VirtPort

There are two parts to the VirtPort application. One portion runs on the host end and is compiled for your specific host operating system. This portion sets up a pseudo-network device driver for the serial

port in question. The other portion is a single program that runs on the PC to provide the PC's serial to network connectivity.

The two programs are:

VIRTPORT.EXE	For the Windows PC
virtport	For the Unix/Linux host. Must be compiled for the correct operating system.

The following text explains the installation and operation of each of these programs. The two programs can be started in either order.

Windows Information (VIRTPORT.EXE)

INSTALLATION

The purchased product consists of just one program, VIRTPORT.EXE. To install, just copy it to your PC's hard disk.

The DEMO product consists of just one program, VIRTPORTDEMO.EXE. To install, just copy it to your PC's hard disk.

OPERATION

The program is a Windows console program, with essentially NO user interface. The command line used to start the program generally contains all the information necessary for VirtPort to run. This command can be entered in any of the usual Windows methods, including a) as part of a BAT file, b) typed in a DOS-prompt, c) typed in Start:Run, d) associated with a desktop icon, etc.

The program is terminated by entering control-C. If the program is run with no parameters, such as:

```
c:\virtport\virtport.exe
```

or

```
c:\virtport\virtportdemo.exe
```

it will assume TCP/IP port 9300, and comm port 1. It will display a Windows dialog box asking you to configure COM1 for baud rate, etc.

Other possible parameters:

/p:<portnum>	specifies which TCP/IP port
/l	logs activity to the screen
/v	verbose - more information than /l
/c:<comm parameters>	sets which comm port, and its parameters

Comm port parameters are in the form

/c:com<n>,<baud>,<databits>,<parity>,<stopbits>,<handshaking>

where <parity> can be "e" for even, "o" for odd, "m" for mark, "s" for space, or "n" for none, and <handshaking> can be "HW" for hardware, "XON" for software (XON/XOFF), or omitted for none.

Examples are:

/c:com1,9600,8,n,1,XON

/c:com3,19200,7,e,1,1.5,HW

Host UNIX/Linux side (virtport)

The virtport program is delivered as a single executable program, typically not compressed or tarred.

The virtport program must be installed by someone with root permissions. For the following steps, establish root permissions by ONE of: a) login as root (or equivalent), b) run 'su', and specify the root password, OR c) run 'sudo' with each of the following commands.

The virtport program must have its ownership and permissions set properly to run. With root privileges, do:

```
chown root virtport
chgrp uucp virtport
chmod 6711 virtport
```

From now on, you do not need root permissions.

Start up virtport with the following command:

```
virtport -i <IP address or hostname> [<options>] [&]
```

This command, with no other options, will connect on port 9300 to the PC at the indicated IP address, and will create a virtual device as

```
/tmp/<username>/virtport.
```

To launch virtport and put it in the background, to allow you (or a shell script) to continue working, include the ampersand ("&") on the end of the command.

To create multiple virtual ports, run multiple instances of virtport, varying the options as needed. Each instance runs one connection to one PC port.

To terminate the virtport session(s), use the 'kill' command to send the process one of the following signals: SIGHUP, SIGINT, SIGQUIT, or SIGTERM.

Options include:

- o Output port number (default 9300)
- v Verbose - lots of output
- s Silent - no output except errors
- l Log - create log file
- w World - make pseudo device accessible by all users
- d Devicename - deprecated; do not use
- D <linkname> see below
- X Allow deleting an old link

The program will assign an available pseudo-tty (that is, a pty) on the Unix/Linux system. This pty is the virtual port that can be used by the controlling program. However, because the virtport takes the first available pty, the particular device assigned will vary from one run to the next. This makes it difficult to coordinate operation with the controlling program. For this reason, virtport creates a soft link from a "linkname" to the assigned pty.

The linkname by default will be based on the username under which virtport is run, as

```
/tmp/<username>/virtport
```

Note that this would allow only one instance of virtport per user. To get around this restriction, use the option

```
-D <linkname>
```

where "<linkname>" is the pathname of a file you want to create. One user can now run multiple instances, each with a different port and linkname.

If the linkname (either default or as specified) already exists, then that could indicate that a previous instance of virtport was terminated abnormally. If that is the case, you can manually remove the old linkname (use the 'rm' command). Or, you can tell virtport that it has permission to delete the old link by using the "-X" option.

If you want the virtual port to be usable by other users on the Unix/Linux system, use the "-w" option.

The Controlling Program

When both ends of virtport are up and running, you can run your "controlling program", the program that needs to communicate with a serial port. In its configuration, where you tell it what serial port to use, you can tell it either the actual pty name assigned by virtport (such as "/dev/tty0"), or the linkname that is linked to it.

Example

In the PC, in a CMD ("DOS") window, I entered:

```
virtport /v
```

It prompted for comm parameters, which I set and OK'd.

On Linux, I entered, as user "ras":

```
virtport -i 204.201.253.34 -v -w
```

It told me it was linking /tmp/ras/virtport to /dev/tty0

Then, from another session:

```
ls > /tmp/ras/virtport
```

Data went through Linux's virtport, through Windows' virtport, to comm port.