COM*t*: the Telnet Modem

Version 1.02

Copyright © 1994, Performance Designs

Overview

Introduction:

Thank you for using **COM*t***, the Telnet Modem for Windows! **COM*t*** is the perfect solution to underpowered Windows Telnet client programs. **COM*t*** allows you to use your favorite Windows communications program (i.e., *Procomm Plus for Windows*, *Crosstalk for Windows*, etc.) to communicate over TCP/IP.

COM*t* does this by installing itself *ahead* of the standard Windows COMM driver (`comm.drv`), and intercepts accesses to the COM ports of your choosing, routing them to TCP/IP. Requests to access other ports are passed on to the original COMM driver, allowing you to continue to use those devices.

COM*t* also emulates a Hayes™-compatible modem, so your existing Windows communications programs will have no trouble interfacing to TCP/IP.

In addition, **COM*t*** can also be used to access network modems, assuming you have modems available via a TCP connection. See the **Auto Initialization** chapter for more details.

System Requirements:

In order to use **COM*t***, you need an IBM-PC compatible system with the following:

- Windows 3.1, Windows for Workgroups 3.1, or Windows for Workgroups 3.11
- 286 processor
- 4 meg RAM
- 256K free disk space
- WINSOCK 1.1-compliant TCP/IP software

Registration:

This program is shareware. It is not free software. If you obtained this copy either from a BBS or a friend, then you are using an UNREGISTERED copy. You may run the program for an evaluation period of one month. After this time if you wish to continue using the program you must register by filling out the Program Registration form and sending in the Registration Fee. This will help guarantee that the program is well supported and enhanced in the future. The shareware copy of **COM*t*** will present a registration reminder dialog on the first usage of a Telnet port.

Currently the price to register **COM*t*** is US $15.95, payable by VISA, MasterCard, check/money-order, or in the SWREG forum of CompuServe. Please see the `register.wri` file for more details.

Getting Technical Help:

Technical support for **COM*t*** is available via email. You can reach us at one of the following email addresses:

CompuServe:	73017,1375
MCI Mail:	572-1706
GEnie:	PERFORM.DES
Internet:	comt@world.std.com

Copyright © 1994, Performance Designs

Installation and Configuration

Installing COM*t*:

COM*t* comes with an automated installation utility. To use this utility, simply copy the distribution files to a floppy or temporary directory, then double-click on the `install.exe` file from the Windows File Manager.

Once INSTALL starts, you will be given the opportunity to read the `readme.txt` file. Then you will be asked to choose a directory to install the **COM*t*** files. Enter a directory name or just press **OK** to keep the default of `c:\comt`. INSTALL will then copy the distribution files to the directory of your choosing, with the exception of the `commt.drv` file, which is always copied to the `\windows\system` directory.

Once all the files have been copied, you will be given the opportunity to allow INSTALL to create Program Manager icons. This is recommended. Once the icons are created, you will be presented with a dialog allowing you to choose which COM port numbers will be assigned to be Telnet ports. It is recommended that you choose any unused ports. You can change these settings after installation using the **COM*t*** configuration program.

This completes the installation. To begin using **COM*t***, simply restart Windows. If the installation fails at any point, please see the `readme.txt` file for instructions on manual installation.

Configuring COM*t*:

You can configure **COM*t*** to provide Telnet connections on up to eight COM ports (COM1-8) by running the **COM*t*** configuration program. Simply double-click the **COMt Config** icon from the Windows Program Manager. You will be then asked to check off which COM port number you wish to be used for Telnet connections. Simply select the desired ports and click **OK**. You can also use this program to uninstall **COM*t*** from your system. Please see the `readme.txt` for complete details on this. Note that any changes you make to the port assignments will not take effect until you restart Windows.

Notes on selectiong COM ports. It is highly recommended that you choose *all* port numbers which you are not using. This gives you the maximum number of simultaneous Telnet connections possible. There does not need to be a physical port for a particular COM device in order to use it for Telnet connections. In fact, you should *not* select ports which *do* have devices attached (i.e., a modem) as you will be unable to access that device while **COM*t*** is installed. If you have a serial mouse connected to a COM port, you will be unable to use that port number for **COM*t***.

Notes on COM5-8. As mentioned earlier, **COM*t*** is able to provide Telnet connections on the ports COM1 through COM8. Unfortunately, because of a limitation in how Windows operates in 386-Enhanced mode, most machines will only be able to use COM1-COM4. You may be able to use the higher port numbers if you have a multi-port board which was provided with a 386 Virtual Device-Driver (usually `serial.386`) which supports port numbers higher than 4.

Notes on COM3-4. On some machines you may run into problems when redirecting COM3 and COM4. Certain terminal programs (*Procomm Plus for Windows* in particular) try to be intelligent about which ports can be accessed without causing hardware conflicts, and may either give unnecessary warnings or refuse to access the port altogether. To overcome this type of problem, add the following lines to the `[386Enh]` section of the `system.ini` file:

```
COM3Irq=11
COM3Base=3E8
```

Copyright © 1994, Performance Designs

```
COM4Irq=15
COM4Base=2E8
```

IMPORTANT: If any of the above lines already exists in the `[386Enh]` section of the `system.ini` file, especially if you have real hardware attached to COM3 and/or COM4, do **not** replace them with the above lines. Doing so may cause you to be unable to access the physical port. If after adding the above lines the problem is not rectified, you may find that it's necessary to adjust the numbers given above

Copyright © 1994, Performance Designs

Using COM*t*

To use **COM*t***, simply instruct your communications program (such as Windows Terminal) to use one of the COM ports you selected to access Telnet. Since **COM*t*** emulates a Hayes™-compatible modem, configure your application to use one of the Hayes Smartmodem, OPTIMA, or ULTRA modems. For the communications parameters, set the connection to any baud rate, with 8 bits, No Parity, and 1 stop bit.

For a phone number, you can specify either the actual IP address of the host to which you wish to connect (i.e., `13.104.22.1`) or the name of the host (i.e., `world.std.com`). Note that by using a hostname, some terminal programs may try to out-smart you by substituting the Touch-Tone™ equivalents for each letter in the hostname (Windows Terminal does this: since *when* have people used their modem to dial 1-800-24PIZZA?). In this case you will need to use the IP address rather than the hostname. Also, if your hostname begins with "t" or "p", you may need to insert either a space or a comma in front of the name in order for it to work properly.

If you wish to connect to a port on the remote host other than the Telnet port (port number 23), simply add the port number in brackets after the IP address. For example, specifying a phone number of `123.456.222.111[25]` will connect you to the SMTP server on that host.

If you want your remote host to automatically recognize a particular terminal emulation, add a section to your modem initialization string which reads `S1000="`*terminal*`"`, where *terminal* is the terminal type you are using. See the *Hayes™-Compatible Modem Emulation* section for more details on S-registers.

Notes on baud rate. You can specify any baud rate for the Telnet port, but you should be aware that **COM*t*** will inform the remote Telnet server of the baud rate you chose. For this reason, you should choose either a very high baud rate (i.e., 38400 or 57600) for TCP connections through a network or the actual baud rate you are using for your SLIP or PPP connection.

Copyright © 1994, Performance Designs

Hayes™-Compatible Modem Emulation:

In order to work as seamlessly as possible with standard modem-oriented programs, **COM*t*** provides a Hayes™-compatible modem emulator. If your application allows to you select a modem type, select one of the Hayes Smartmodem, OPTIMA, or ULTRA modems. This will ensure that the program initializes and dials the modem emulator properly.

The modem emulator supports the standard "AT" command set. Most AT commands are ignored, as they only have relevance when controlling a real modem. The AT commands which **are** acted upon are listed in the next section.

Modem Emulator AT Command Set:

D -- Dial Command

This command instructs the modem emulator to attempt to establish a connection to the named-host or IP address following the **D** command. This works similarly to a modem, except that a network address is specified instead of a phone number. The "T" and "P" (tone or pulse) modifiers are allowed to follow the **D** command, and they are simply ignored. The modem emulator attempts to connect to the specified host, returning a `CONNECT` response string on success or `NO CARRIER` upon failure. If the connection fails, an additional error message beginning with `WINSOCK:` is output, providing more detail as to why the connection failed. As with a normal modem, if you press any key while the connection attempt is in progress, the connection will be aborted and `NO CARRIER` will be output.

E0 or E1 -- Command-mode echo

This command enables or disables character echoing while the modem emulator is in "command mode" (i.e., accepting "AT" commands). **E1** enables echo and **E0** disables echo. The default is **E1**.

F0 or F1 -- Online echo

This command enables or disables character echoing while the modem emulator is in "online mode" (i.e., actually connected to a remote host). **F1** enables echo and **F0** disables echo. The default is **F0**.

H0 -- Go off-hook

This command is only valid when the modem emulator has established a connection, and causes the emulator to close the current connection.

O -- Return to online mode

This command is only valid when the modem emulator has established a connection, and causes the emulator to exit "command mode" and returns control of the serial line to the current Telnet connection.

Sr=n -- Write to an S-register

This command writes a value *n* into the S-register numbered *r*. The list of S-registers supported is listed later. Any write to an unsupported S-register is simply ignored.

Sr? -- Query an S-register

Copyright © 1994, Performance Designs

This command outpus the current value of the S-register numbered *r*. The list of S-registers supported is listed later. Any write to an unsupported S-register causes the word ERROR to be output and parsing of the current command line is terminated.

W0, W1, W2 -- Negotiation progress messages

In addition to the CONNECT response string, this command can be used to enable the CARRIER, PROTOCOL, and COMPRESSION messages that are available with some MNP or v.42 modems. **W0** and **W2** disables the extended progress messages, and **W1** enables the CARRIER and PROTOCOL messages. Note that any value other than zero in **S95** will override this setting. Also note that since no compression or transmission protocols are actually being used, they are always reported as NONE. The default is **W0**.

Z -- Reset

This command resets the modem emulator to use the default settings, and will close any currently open connection.

&C0, &C1, &C2 -- Data Carrier Detect options

This command controls how the Data Carrier Detect (DCD) line is reported by the modem emulator. **&C0** causes the emulator to always report the DCD line as ON. **&C1** causes the emulator to report the DCD line as OFF while there is no connection, is switched to ON immediately after the CONNECT message and OFF again when the connection is lost. **&C2** causes the emulator to report the DCD line as ON while there is no connection, and as OFF while a connection is being established, then ON again once the connection is made. The default is **&C1**.

&D0, &D1, &D2, &D3 -- Data Terminal Ready options

The command controls how control of the Data Terminal Ready (DTR) line by an application is interpreted by the modem emulator. **&D0** causes the emulator to ignore the DTR line. **&D1** causes an ON-to-OFF transition of the DTR line to simply put the emulator in command mode while maintaining any currently open conenction. **&D2** or **&D3** causes an ON-to-OFF transition of the DTR line to close any currently open connection (issuing a NO CARRIER response if one was open) and puts the emulator in comand mode. The default is **&D0**.

&F -- Factory Defaults

This command acts exactly like the "Z" command.

&S0, &S1, &S2 -- Data Set Ready options

This command controls how the Data Set Ready (DSR) line is reported by the modem emulator. **&S0** causes the emulator to always report the DSR line as ON. **&S1** causes the emulator to report the DSR line as OFF while there is no connection opened, switches it to ON when attempting to open a connection, then OFF again once the connection is closed. **&S2** causes the emulator to report the DSR line as OFF while there is no connection opened, switches it to ON when a connection is opened, then OFF again once the connection is closed. The default is **&S0**.

&V -- View current settings

This command displays the current settings being uses by the modem emulator. A sample output (using the default settings) is shown below:

```
ACTIVE PROFILE:
```

Copyright © 1994, Performance Designs

```
E1 F1 W0 &C1 &D0 &S0
S02:043 S03:013 S04:010 S05:008 S12:050 S95:000
S1000:000
S1001:002 S1002:000 S1003:001 S1004:001 S1005:003
```

Modem Emulator S-registers:

S2 -- Escape Sequence Character

This register holds the ASCII value of the escape character used in the escape sequence. The escape sequence is defined by a period of silence (as defined in S12) followed by three escape characters followed by another period of silence. This causes the modem emulator to switch from online mode to command mode while a connection is open. The default is 43 (the "+" character).

S3 -- Carriage Return Character

This register holds the ASCII value of the character which is both understood and output for a carriage return. The default is 13.

S4 -- Line Feed Character

This register holds the ASCII value of the character which is both understood and output for a line feed. The default is 10.

S5 -- Backspace Character

This register holds the ASCII value of the character which is both understood and output for a backspace. The default is 8.

S12 -- Escape Sequence Gaurd Time

This register holds the value of the silence period required when issuing an escape sequence, in 1/50th of a second increments. The default value is 50 (1 second).

S95 -- Negotiation progress messages

This register, when set to anything other than zero, overrides the W setting in how the emulator reports connection progress. Setting bit 2 of the register enables the `CARRIER` message. Setting bit 3 of the register enables the `PROTOCOL` message. Setting bit 5 of the register enables the `COMPRESSION` message. The default is 0.

Modem Emulator extended S-registers:

For the purpose of setting Telnet-specific options, the modem emulator supports an extended set of S-registers, numbered 1000 through 1005.

S1000 -- Terminal Type

This register is somewhat unique in that it can accept both a numeric *and* a string value. By setting this register to a value other than zero, any Terminal-Type query by the remote Telnet server is answered with the current value of this register. By setting this register to a string value (i.e., `ATS1000="vt100"`) , the verbatim string is reported. By setting this register to a numeric value (i.e., `ATS1000=5`), the numeric value (presumably a well-

Copyright © 1994, Performance Designs

known terminal id) is reported. The default is 0.

S1001 -- Interpret IAC characters

The Telnet protocol is based on an escape character (ASCII 255) known as the IAC character. This register controls whether **COM*t*** will interpret and deliver these escape sequences. Setting this register to 0 causes the client to treat the IAC character as any other. Setting this register to 1 causes **COM*t*** to always interpret IAC. Setting this register to 2 causes **COM*t*** to interpret/ignore based upon the requested port number. If a connection to port 23 is made (the default in the **D** command, and the standard Telnet server port), then IAC escapes are interpreted. If a connection to any other port number is made (by using `[nn]` in the **D** command), then IAC characters are treated like any other. The default is 2.

S1002 -- Request Binary Connection

The Telnet connection can be used either in Terminal and Binary modes. This register controls whether or not a Binary connection is requested by **COM*t***. Setting this register to 0 causes **COM*t*** to *never* request a Binary connection. Setting this register to 1 causes **COM*t*** to *always* request a Binary connection. Setting this register to 2 causes **COM*t*** to request a Binary connection only if the port was opened using a byte length of 8 bits. The default is 0.

S1003 -- Telnet Echo

The Telnet server can be operated in either Echo and Non-Echo modes. This register controls whether or not an Echo mode connection is requested by **COM*t***. Setting this register to 0 causes **COM*t*** to *never* request an Echo mode connection. Setting this register to 1 causes **COM*t*** to *always* request an Echo mode connecton. The default is 1.

S1004 -- Verbose Connection Failure

This register controls whether or not the additional `WINSOCK:` error message is reported before the `NO CARRIER` response. Setting this register to 0 disables the `WINSOCK` message, and setting it to 1 enables the message. The default is 1.

S1005 -- Carriage-Return Padding

The Telnet protocol specifies that all Carriage-Return characters (ASCII 13) which are sent without a Line-Feed character (ASCII 10) should be sent as "CR NUL" (ASCII 13 followed by ASCII 0). Setting this register to 1 causes **COM*t*** to strip away any NUL character received which follows a Carriage-Return. Setting this register to 2 causes **COM*t*** to add NUL characters to any outbound Carriage-Return which is not directly followed by a Line-Feed. Setting this register to 3 enables both inbound *and* outbound translations, and setting it to 0 disables both. Note that S1001 will override this setting, such that if IAC characters are not being processed, neither will CarriageReturn characters be processed. The default is 3.

Copyright © 1994, Performance Designs

Auto Initialization:

COM*t* supports an automatic initialization feature which allows you to specify a string to send to the modem emulator upon opening the port. This can be used as a convenience feature, such that the modem emulator can be always initialized to a known state each time it is opened. In addition, this feature can be used as the basis for accessing remote network modems.

For example, to automatically initialize the modem emulator when COM3 is opened, add a line to the `[COMt]` section of `system.ini` which reads:

```
COM3init=AT &C1 &D2 S1000="vt100"
```

That would enable DCD and DTR handling (as described in the previous section), and set the reported terminal to be "vt100" every time an application opens COM3.

To illustrate using **COM*t*** to access networked modems, let's assume that on a remote server (IP address 123.456.789.111) there is a modem which can be accessed by connecting to port 4006. To cause an application to be automatically connected to that modem every time COM3 is opened, add a line to the `[COMt]` section of `system.ini` which reads:

```
COM3init=AT &C1 &D0 S2=0 D:123.456.789.111[4006]
```

Note the colon after the "D" command. This is very important, as it causes **COM*t*** to "hold" the port until the connection is made, making the connection transparent to the application. Also note the "S2=0", which effectively disables local interpretation of the escape sequence so that an escape sequence can be used for the remote modem.

Please note: because the application will be unable to see the true status of the network modem's status lines, and will be unable to manipulate that modem's control lines, some applications may not work correctly when used in conjunction with this feature.

Copyright © 1994, Performance Designs

Application Notes:

Using WinCIM:

COM*t* can be made to work with CompuServe's WinCIM, although you either need WinCIM version 1.3 or the latest support files. Of key importance is the ability to dial using the network setting of "Internet". If that option is not available in your copy of WinCIM, you need to either upgrade to version 1.3 or obtain the latest support files from CompuServe.

To use **COM*t*** with WinCIM, change the Session Settings as follows:

```
Phone: compuserve.com
Baud Rate: 9600
Network: Internet
Connector: (one of the Telnet Modems)
```

In addition, set the Modem Control String as follows:

```
Initialize: AT&F &C1 &D2 S1005=1 S1002=1^M
```

Transferring Binary Files:

Transferring binary files through a Telnet connection can be extremely problematic, as the interpretation of Binary Mode and Carriage-Return padding is quite varied among Telnet implementations. To get binary file transfer to work properly you may need to experiment with S1002 and S1005 to find the best settings. On at least one SPARC-based telnet server, S1002=1 and S1005=0 works best with binary files, although many files will transfer fine with S1002=0 and S1005=3. As noted above, S1005=1 and S1002=1 works best for the HMI protocol used by WinCIM. You also might try switching to YMODEM protocol rather than ZMODEM protocol, as in some cases better results can be had using YMODEM.

Using Smartcom for Windows:

COM*t* can be made to work with Hayes' *Smartcom for Windows*, and to do so you need to check the "Use reduced command set" box in the Modem Settings. Failure to do so will prevent dialing using **COM*t***.

Using Reflection:

Reflection uses its own datacom driver, bypassing **COM*t*** and the Windows COMM driver. In order to get **COM*t*** to work with *Reflection*, issue the following command in the *Reflection* Command Window.:

```
SET ENHANCED-SERIAL-DATACOMM NO
```

Copyright © 1994, Performance Designs