Contents

Click a topic below for information on the features and functions of *QmodemPro for Windows.* Or use the **Search** Button above this text to search for a particular topic.

How-To . . .   A selection of topics of interest to new users and experienced users alike. How to upload or download a file, send a fax, store information in the phonebook, and much more.

Menu Commands   A listing of all menu commands, including all *QmodemPro for Windows* functions.

Dialog Boxes   A description of every dialog box, including options, defaults, fully cross-referenced for your convenience.

Terminal Emulations   Keyboard layouts for all terminal emulations supported by *QmodemPro for Windows*.

Troubleshooting   Look here for help when things go wrong. Helpful hints for solving some of the more common problems, and information on how to get further assistance when you need it.

SLIQ Script Language Interface for QmodemPro   A complete guide to the *QmodemPro for Windows* script language. Use SLIQ to automate your logins, or perform sophisticated programming tasks.

About MSI   More products of interest to modem users, from Mustang Software, Inc.

Last Minute Changes   Late-breaking news -- last minute feature additions, and things that arent in the manuals.

Hints: You can page through help topics by using the **Browse** buttons (the ones labelled << and >>. Try the **History** and **Back** buttons to return to a previous topic, and the **Contents** button to see this screen again. For more information on how to use Windows Help, press your **F1** key.

To see help about a specific menu command while youre running *QmodemPro for Windows,* hold down the left mouse button on the menu command youre interested in,

and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

Index

T

U

V

Last Minute Changes

The following list describes changes to the program that were made after the manuals were printed. In all cases the on-line help file (QMWIN.HLP) has been updated in the appropriate places to reflect these changes.

The Registration Card is bound inside the back cover rather than inside the front cover. Please complete and mail this card right away to ensure you are notified of program updates and special offers.

Program Additions:

The file DEFAULT.EMU is provided in case you want to restore your QMWIN.EMU to the default state. Simply copy DEFAULT.EMU to QMWIN.EMU to restore the original emulation settings.

Added confirmation on Delete Fax and Delete Group.

Pressing **[Enter]** in Phonebook will now dial marked entries. **[Alt][Enter]** will edit highlighted entries in the phonebook.

Added baudrate of 230,400 to the device settings.

BMP (Windows Bitmap) files are displayed in the GIF viewer as they are downloaded. GIF viewer allows you to delete a GIF file after downloading.

The script DOWNLOAD command now goes to the default download directory if a blank string is specified for the directory.

Added support for ANSI music. This option can be turned on from the Options/Emulations dialog, Options button.

Added a script called SERVICES.SCR that automates connectivity to CompuServe, GEnie and MCI Mail. This script should be used to set up and automatically configure your dialing directory entries for these services.

Added support for international keystrokes in the Options|Desktop dialog. The new toggle, Accept international keystrokes turns on an AnsiToOem conversion on typed characters. This same conversion operates on macro keys, keyboard definition keys, and translate table entries if the international option is enabled. This allows certain characters typed on international keyboards to display correctly.

Alternative code pages are supported -- if *QmodemPro for Windows* finds a code page in your environment, it will search for a QMWFONT.xxx where xxx is the code page, otherwise QMWFONT.FON will be loaded. An alternative character set for code page 865 (Scandinavia) is included with *QmodemPro for Windows.*

Pressing F1 while a menu is pulled down will display help for that item.

Added a Window menu to the script debugger menu for changing window views.

Added a warning dialog when trying to close QmodemPro while still on-line.

Added a Clear button to the sounds configuration dialog.

Added Vidtex emulation for use when connected to CompuServe.

Added confirmation on deletion of devices.

Added support for auto display of .GIF images while on-line with CompuServe, for example the weather images sent in GO WEATHER.

Added new script error messages:

"Invalid opcode"

"Unexpected end of file"

"STOP encountered"

"RETURN without GOSUB"

"Math error"

"Array subscript out of bounds"

"Library not found"

"Function not found"

"Invalid file number"

"Error opening file"

"Path not found"

"Error renaming file"

"Communications timeout"

"WHEN ... GOTO executed from noncurrent scope"

Added new script commands:

function ADDLFTOCR [on/off]

function CONFIGCAPTUREFILE

function CONFIGDOWNLOADPATH

function CONFIGLOGFILE

function CONFIGSCRIPTPATH

function CONFIGSCROLLBACKFILE

function CONFIGTRAPFILE

function [CONFIGUPLOADPATH](#)

function [DOORWAY](#) [on/off]

function [SCROLLBACKRECORD](#) [on/off]

function [STRIPHIBIT](#) [on/off]

statement [AUTOANSWER](#) on/off

statement [EDITFILE](#) filename

statement [MOUSECLICK](#)

statement [PRINTER](#) on/off

statement [RESETEMULATION](#)

statement [RESETMODEM](#)

statement [SETDTR](#) on/offf

statement [SCROLLBACK](#) on/off

statement [SPLITSCREEN](#) on/off

statement [TRAPSCREEN](#) filename

statement [VIEWFILE](#) filename

statement [VIEWGIF](#) filename

statement [XONXOFF](#) on/off

Program Changes:

When [Quicklearning](#) a script, the RunScript button is changed to EndLearn which will end the Quicklearn.

The [Emulation](#) setup dialog disables font and color selections on the RIP emulation since the color is dynamic and the font is a fixed EGA specification.

All [DOS Shells](#) now use the COMSPEC setting to launch DOS instead of DOSPRMPT.PIF. This change was necessary to work properly with other operating environments including OS/2.

The [Phonebook](#) now prompts you if it contains phonebook entries that have their device field set to a non-existent device. At this point you can have the it change the entries to the default device automatically if desired.

Errata

Some of the screen snapshots in the printed manuals may differ in minor detail from the screens which appear in the actual program. In all cases, these variations will not affect the operation of *QmodemPro for Windows*. For up-to-date information on a screen, use the on-line help.

The following errors and omissions in the printed manuals have been corrected in the on-line help file where applicable:

User Guide

Page 93: The Macro System Command @MANUALDIAL was inadvertently left out of the list of commands.

Page 131: Add Entry, "The Phonebook" The reference to Chapter 4 should be Chapter 3.

Page 139: Break, "Fine Tuning" The reference to Chapter 4 should be Chapter 3.

Page 148: Device Connection. The default for Full/Half Duplex is actually "Full" not "Half".

Page 153: Device Modem, "Special Control Characters" The reference to Chapter 5 should be Chapter 4.

Page 167: Edit a File, "The Editor" The reference to Chapter 5 should be Chapter 4.

Script Guide

Page 65, DIALOG Statement line 13: [integer-field LISTBOX ... should be [integer-field AS LISTBOX ...

About MSI

Mustang Software, Inc. offers a range of products for your telecommunication needs. All these products are available through your usual retail and mail order channels.

Wildcat! BBS

Since Mustang Software, Inc. first introduced *Wildcat!* in 1986, it has grown to become the world's best selling Bulletin Board System (BBS) software.

Much of this success is due to *Wildcat!*'s flexibility and economy. Four versions of *Wildcat!* provide the power and features you need, from an economical single-line installation to a sophisticated LAN-capable multi-user system.

Wildcat! Single-Line is an inexpensive solution for those wishing to begin with a single telephone line and modem. Lower price doesn't mean lack of power with *Wildcat!* Single-Line. In fact, with the exception of multi-user chat, *Wildcat!* Single-Line contains exactly the same features as our Multi-Line, Professional and IM versions.

Wildcat! Multi-Line supports up to ten concurrent users in any combination of LAN or modem connections. Up to four telephone lines can be connected to a single computer using DESQview, while connecting multiple PCs to a Local Area Network provides for up to 10 simultaneous connections. Except for the number of concurrent users it supports, *Wildcat!* Multi-Line has all the features and functions of *Wildcat!* Professional.

Wildcat! Professional has all the features of *Wildcat!* Multi-Line, but supports up to 250 concurrent users by connecting multiple PCs to a Local Area Network. Up to four telephone lines are supported on a single PC using DESQview and *Wildcat!* Professional.

Wildcat! IM (Intelligent Multi-Port) was the first BBS software to offer DigiCHANNEL support. Using *Wildcat!* IM, you can support up to eight telephone lines on one computer and up to 250 total concurrent users via a LAN.

wcPRO -- The PRO! Series Utilities

This integrated utility program offers extended database processing and graphic analysis to the BBS sysop. It is used to manipulate and evaluate many areas of BBS operation. The series includes packages for *Wildcat!* database management, LAN node monitoring, and a Fax-on-demand system for your BBS.

wcMHS -- Netware MHS gateway for Wildcat! BBS

This gateway is designed to work with Novell's Message Handling Service (MHS). It allows *Wildcat!* BBS messages to be redirected to any other MHS compatible application. *wcMHS* can also be used to link *Wildcat!* BBSs together, echoing messages via MHS direct or through the CompuServe MHS Store and Forward Hub.

wcUUCP -- Internet E-Mail and Newsgroup gateway for Wildcat! BBS

The latest addition to the *Wildcat!* family sends and delivers E-Mail and Newsgroup mail between your *Wildcat!* BBS and your UUCP host. Get into the fast lane on the electronic

data highway, and start talking to the world!

QmodemPro for DOS

For more than nine years the DOS version of *Qmodem* has been the first choice of well over a quarter of a million pioneering communication enthusiasts worldwide.

The latest release of Qmodem, *QmodemPro for DOS,* is available in a package designed for you -- whether you're just discovering the on-line community, or you're a discriminating power user who demands a high performance, thoroughly professional package.

Off-Line Xpress

.QWK-compatible mail packet reader for Tomcat, QMail, and more. Reading mail from BBSs has never been easier. OLX can read packets generated by any .QWK compatible mail door used by most BBS, and generates reply packets in the .QWK format which can be uploaded back to the originating BBS. Included FREE with *QmodemPro for DOS,* or available separately for only $40.

Procedures

This is a how-to list for some of the more common functions of *QmodemPro for Windows*.

[Changing the appearance of the terminal screen](#)

[Character translation](#)

[Command line options](#)

[Connecting and logging on](#)

[Customizing the desktop](#)

[Devices](#)

[Direct Connection via Serial Cable](#)

[Emulations](#)

[Emulation KeyMaps](#)

[Fine tuning](#)

[How do I send a file to my buddy?](#)

[Marking and Pasting text on the screen](#)

[Opening a DOS Window](#)

[Other communication devices](#)

[Programmable Keys](#)

[Quick Learning Scripts](#)

[Recording your Session](#)

[Sending and Receiving Fax Messages](#)

[Special Control characters](#)

[Starting QmodemPro for Windows](#)

[The Editor](#)

[The Phonebook](#)

[Uploading and Downloading files](#)

[Using a network modem with QmodemPro for Windows](#)

[Viewing a file](#)

[What to do with compressed files](#)

Starting QmodemPro for Windows

There are several ways to start *QmodemPro for Windows.*

From the Program Manager, *QmodemPro for Windows* Program Group

Find the *QmodemPro for Windows* program group in your Program Manager window, and double-click on the *QmodemPro for Windows* icon.

From the Program Manager, File menu

Select the **File** menu, and click on the **Run ...** command. In the text window that pops up, type

`QMWIN` **[Enter]**

From the Program Manager, Startup Group

Drag the *QmodemPro for Windows* Icon into the **Startup Group** for automatic execution whenever you start Windows.

You can also pass parameters on the command line to start *QmodemPro for Windows* with a particular phonebook file loaded, or to automatically dial a number or group of numbers, or execute a script. If you're interested in reading more about startup options, please refer to the **Command Line Options** topic.

The Phonebook

QmodemPro for Windows's **Phonebook**, or dialing directory, is a window organized in much the same way as your personal address book. Like your address book, it can contain information including the system name, telephone numbers, your logon name, password and communication parameters. You can add a note file to each entry, to record more detailed information and comments. And you can mark and organize entries into groups, for multiple dialing sessions.

Each Phonebook stores up to 4096 entries, and you can have an unlimited number of Phonebook files. The default phonebook filename is QMWIN.PHN. The filename and path of the current phonebook is printed on the Title Bar at the top of the Phonebook window.

The directory is organized in rows and columns, with information about each entry displayed horizontally across the screen. The first field, Number, is the entry's position on the list.

Other fields contain information about the service; the name and phone number for the service, your User ID, the name of a note file associated with the service, the date and time of your last connection, and so on. Use the scroll bars at the bottom of the window to view these fields.

[Adding or Revising an entry](#)

[Deleting an entry](#)

[Dialing an entry](#)

[Finding an entry](#)

[Group dialing](#)

[Loading a different Phonebook](#)

[Manual dialing](#)

[Packing the Phonebook](#)

[Printing the Phonebook](#)

[Sorting the Phonebook](#)

[Tagging entries in the Phonebook](#)

Adding or Revising an entry

Use the **Add** button on the tool bar to add new directory entries, and the **Revise** button to edit an existing entry. The <u>**Revise Phonebook Entry**</u> dialog box will pop up, allowing you to edit or add information about the selected Phonebook entry.

Filling in System Information

System name is the first item. This is the name of the service you call -- for instance "MSI HQ BBS" or "CompuServe". When you have entered the system name, press your **[Tab]** key to move to the next field.

The next five entries are for telephone numbers. If the system you call has only one telephone number, enter it on the first line.

Many multi-user services have more than one telephone number -- in this case use the **Alt. #** fields to enter additional numbers. When you dial an entry with multiple phone numbers, *QmodemPro for Windows* will call each one in succession until it connects.

Retries tells *QmodemPro for Windows* how many times to redial an entry if it does not connect on the first call. Set this value to zero (the default) to redial an entry continuously, or until you manually abort the redial.

To change the number of **Retries**, type the new value into the text box. Use this option with discretion -- your local telephone company may have rules limiting the number of times you may dial a number within a certain period.

User ID is the next field. Enter the name or logon ID you use to connect with the service you are calling. The information in this field can be sent with a <u>@USERID</u> macro command.

The **Password** field is self-explanatory. Enter the password you use to connect with this service. This information stays hidden until you select the Password field to edit, and your password can be sent with a <u>@PASSWORD</u> macro command.

If you store your password here, you can recall it easily during logon with a keyboard macro or a logon script. This saves you the trouble of remembering many different passwords, and providing a measure of security for your on-line activities.

You should always take care to prevent unauthorized use of your logon accounts. Use a different password on every system you call. Computer security experts recommend using non-obvious passwords. Your own name or initials, or the name or initials of your spouse, children or pets are all poor choices for passwords -- while they are easy for you to remember, they are also easy for a potential intruder to guess.

The next field is **Note.** *QmodemPro for Windows* allows you to attach a file to a phonebook entry. This can be any kind of file, and the same file may be attached to more than one phonebook entry. The name and path of the attached file are stored in this field.

This field has an **Edit** button, allowing you to select a Note file from a Windows Common Dialog Box.

QmodemPro for Windows uses the Note file in a special way. When you attach a note file to a phonebook entry and use the <u>**Find**</u> command on the Phonebook menu to locate and tag entries for dialing, the contents of the note file are searched along with the text in the

System, UserID, Password and other text fields for matching text.

Macros is the name of the Macro file associated with this entry. The Macro file references which set of macro assignments will be used for the buttons below the terminal window when a particular phonebook entry is dialed. This option allows you to customize your keyboard's operation for each system.

This field has an **Edit** button, allowing you to select a Macro file from a Windows Common Dialog Box.

Script is the name of the script file *QmodemPro for Windows* will execute whenever you dial and connect with this Phonebook entry. If the script file does not exist on disk, *QmodemPro for Windows* will create one for you automatically, in QuickLearn mode.

This field has an **Edit** button, allowing you to select a Script file from a Windows Common Dialog Box.

RIP Icons is the path for RIP (Remote Imaging Protocol) icons associated with this phonebook entry. Bulletin Board Systems making use of the RIPscrip terminal emulation may have their own set of icons, which are stored on your hard drive, and which are displayed automatically when you view the BBS menus and screens.

Since different Bulletin Board Systems may use similar filenames for RIP icons, you may wish to store icons associated with a particular BBS in a separate directory. An entry here will override the default RIP Icon path specified in the **Options/Files/Path Definitions** dialog box.

This field has an **Edit** button, allowing you to select the RIP Icons path from a Windows Common Dialog Box.

The **Device to use** list box lets you select the hardware **device** from which the call should be made. This option allows you to use a variety of devices, including one or more modems in your desktop PC, including Fax modems. Other devices such as network modems are also supported, providing the proper drivers are installed in Windows.

The next field is **Terminal Emulation.** Your PC emulates, or imitates, a terminal during a communications session, and many different types of terminals are supported by *QmodemPro for Windows*. **Terminal Emulation** controls the way your computer sends, interprets and receives keyboard and video information from the host computer. This option allows you to select the terminal emulation you wish to use for this system. In most cases, the best choice is **ANSI**.

On host systems that support RIP (Remote Imaging Protocol) graphics, try the **RIPscrip** emulation. On mini or mainframe systems, ask the system administrator for guidance on which terminal emulation is best for the applications you want to run.

Next, select the default **file transfer protocol** you wish to use with this system. In most cases, the best choice is Zmodem. You'll find a more detailed discussion of transfer protocols later on in this chapter.

The next item is **Fax Device Only.** Toggle this item **on** if you want to send Faxes to this phonebook entry using *QmodemPro for Windows* and your Fax modem.

Next, if you like, you can attach a Windows **Icon** to a phonebook entry. A number of sample icons have been already been installed in your *Qmodem for Windows* home directory. There are dozens of others in your Windows directory, stored in a file named

BBSICONS.DLL. You can view your phonebook as a collection of icons, as text only (the default), or as icons plus text, by changing the **Phonebook/View** options.

QmodemPro for Windows keeps track of **statistics** for each entry: the number of calls to a system, plus the date and time of your last call, the number of uploads and downloads. The **Clear** button will reset the statistics for this system to zero.

Deleting an entry

The **Delete** button will remove tagged entries or pending Fax messages from your phonebook. A confirmation window will pop up before the entry or entries are deleted. Be careful with this option -- once an entry has been deleted, it is gone, and can't be recovered.

Dialing an entry

Use the **Dial** button on the tool bar to begin dialing selected entries in numerical order. The Redialer will cycle through the tagged entries until a connection is made. Press **[Esc]** to halt the process.

When *QmodemPro for Windows* connects, the screen will clear. *QmodemPro for Windows* plays the sound you have selected for a successful connection (if you have turned on **Options/Sounds** and you have a sound card or a sound driver for your PC speaker), goes into terminal mode and allows you to log into the remote system.

Finding an entry

The **Find** command scans the phonebook for a specific text string. Entry Names, Phone Numbers, UserID, Password, Script, Macro, and RIP fields are searched starting with the current entry. Searching can be done forward or backward from the current entry.. Any file linked to the Note field is also searched.

The find dialog box has a data entry field for the Text to Find, and three option buttons. The option buttons are **Find Next**, which searches for each matching entry; **Tag All** which automatically finds and tags all matching entries; and **Close**, which returns to the Phonebook dialog box.

Group dialing

Selected phonebook entries can be saved as groups. These groups can be used for a variety of purposes, for instance you may select a group of entries and dial them immediately, or tag them for further processing from the Phonebook window.

To create a group, select the **Group** button on the Tool bar. This pops up a dialog box with a window listing groups, and several control buttons. The first two buttons, **Add** and **Update**, operate similarly, and are used to create and edit groups.

Creating a group

Use the **Add** button to create a new group. Use the Group Name field to store the name of the group. Use a descriptive name such as "Mailrun" or "Games" or "Local BBSs" or anything else that describes what these systems have in common. Note that you can add the same Phonebook entry to more than one group.

The **Continuous Dial** toggle in the next field causes *QmodemPro for Windows* to dial each entry until you have connected with each system on the list.

The next two fields are selection lists. The left window lists all your Phonebook entries, while the window on the right side lists the entries you have selected for your Group. To select an entry, highlight the system in the **Phonebook Entries** list, then press **[Enter]**, or select the **Add** control button. That name will then be copied automatically to the **Selected for Group** list. You can also double-click to add an entry.

To remove an entry from your Group, go to the Selected for Group window, highlight the entry you wish to delete, and press **[Enter]** or select the **Remove** control button. To remove ALL entries from a group, select the **Clear** control button. You can also double-click to remove an entry.

When you have finished building or revising your group list, select **OK** to save. You will then be returned to the Group Options dialog box.

Revising a group

To revise a group, highlight the group you wish to edit, then select the **Revise** button. The Revise dialog box operates in exactly the same way as the Add dialog box for adding and removing entries in your group.

Deleting a group

To delete an entire group, highlight the group or groups you wish to remove, and select the **Delete** button.

Tagging a group

The **Tag** button highlights (tags) all group entries in the current phonebook. When the entries are tagged, you can go back to the Phonebook window and perform other operations on the tagged entries, for instance dial, delete, or revise.

Dialing a group

Use the **Dial** button to begin dialing all members of the group. The Dialer dialog box will appear, showing the progress of the current call.

Loading a different Phonebook

The **File/Open** menu command allows you to load a different phonebook from disk. You can have as many Phonebook files as you want, stored anywhere on your hard drive. You can create additional Phonebooks by opening the Phonebook and selecting **File/Open.** Type the name of the phonebook you'd like to create (the extension .PHN is optional), and click **OK.**

Manual dialing

The **Manual Dial** button allows you dial a number that is not in your Phonebook. Manual dial commands may contain up to 40 characters, and *QmodemPro for Windows* does not check the data for validity. The data typed in the Phone Number field will be passed to the modem as if it was attached to a real entry. Any Prefix modifier will be substituted accordingly.

Packing the Phonebook

You may notice that when you delete entries from the Phonebook, the size of the file on disk does not decrease. This is because deleted records are not automatically removed from the file. Use the **Pack** command on the **File** menu to remove deleted record space from your Phonebook file, and shrink the file back down to a more manageable size.

Printing the Phonebook

The **Print** button allows you to copy selected information from your phonebook to a printer or to a disk file.

A dialog box with a list box allows you to select any combination of fields in your Phonebook to print. Highlight each field you would like to print, and use the **Add** button to add them to the list on the right side of the dialog box. Use the **Up** and **Down** buttons to change the order of fields in your list.

When you are satisfied with your selections, click the **Print** button. This pops up the standard **Print** dialog box.

Click the **Setup** button if you want to change the default printer, or click the **Print to File** check box to redirect the output to a DOS text file.

The output from the print phonebook command is one record per line, and is not formatted for line length. You should take your printer's maximum line length into consideration when selecting fields to output.

Sorting the Phonebook

This option allows you to **<u>Sort</u>** the current phonebook. Any of the fields can be used as the primary sort key, and you can sort in ascending or descending order.

For example, if you wanted to sort the phonebook so that the entry with the highest number of calls comes first, and the least called entry at the end, you would select Total Connects from the Phonebook fields list, add it to the Sort fields list, select the Descending button, then click **OK**. As with the **Print Phonebook** dialog box, the **Up** and **Down** buttons can be used to rearrange entries on your list.

Tagging entries in the Phonebook

Tagging is a way to build a queue of entries to dial, revise, copy, cut or delete. There are several ways to select and tag phonebook entries, all following the standard Windows procedure for selecting single and multiple entries.

With a Mouse

Click the first item you want to select, and the drag the cursor to the last item you want to select.

To cancel the selection click on any item after doing the above.

Or click the first item you want to select, hold down the **[Shift]** key, and click on the last item you want to select. This will select the first and last items, and all items between.

To select non-sequential items, hold down the **[Ctrl]** key and click each item you want to select. To cancel a selection hold down the **[Ctrl]** key and click the selection again.

From the Keyboard

Use the up and down arrows to select an individual item. To select more than one item hold the **[Shift]** key down while pressing the up and down arrows to select a group of entries.

To select all items from the current item to the beginning of the phonebook hold down the **[Ctrl]** key and the **[Shift]** key and press **[Home]**. To select all items from the current item to the end of the phonebook hold down the **[Ctrl]** key and the **[Shift]** key and press **[End]**.

To select multiple entries activate **Add mode** by pressing **[Shift][F8]**. The focus indicator (the dotted box) can then be moved independently of the selection highlight, allowing additional entries to be highlighted by pressing **[Space]**. When the desired entries are selected, turn **off** Add mode by pressing **[Shift][F8]** again.

You may then return to the Phonebook window to further process the tagged entries. Note that this operation first un-tags any tagged entries in the Phonebook, then applies tags to the entries in the group.

Connecting and logging on

Once you've dialed the host system you want to connect with, the next step is to log on. Logging on means identifying yourself to the host computer by typing your name or user ID, and your password.

If the system you call is connected through a telephone network to a local access number, you may first need to hit the **[Enter]** key a couple of times to wake up the network, then identify the service you wish to reach, for example "CIS" for CompuServe Information Service.

Most host systems, whether they are Bulletin Board Systems, On-line Services, or private computer systems, offer some kind of prompt or menu system to give callers an idea what kind of input the host computer is expecting. If you're lost, type a question mark character or the word "Help". Most services offer at least some level of on-line help.

For a detailed, step by step description of how to log onto a BBS, enter a message, and download a file, please review the tutorial in Chapter 2 of the manual. While the commands and prompts might vary somewhat from one host system to another, the same concepts and procedures will apply to most computers you call.

In most cases, the logon will proceed smoothly. If it doesn't, for instance if your terminal window fills up with garbled characters instead of the text you were expecting, refer to the appendix at the end of this book, in the section on troubleshooting and problem solving. If you can't make sense of what you're seeing on the screen, use the **Terminal/Hangup** command or click the hang-up button in the Tool Bar to disconnect, then review your settings and try again.

Marking and Pasting text on the screen

Using the mouse, you can mark text on your screen during an on-line session, and send it to the modem. This is a useful way to grab a file name from a listing, then download the file, without ever having to type the name again.

To mark text on the screen, move the mouse pointer to the beginning of the text you want to mark. Hold down the left mouse button and drag the mouse to the end of the block of text. The marked text will be highlighted in a contrasting color.

To send the marked text to the modem, click the right mouse button.

You can also mark text on your screen and copy it to the clipboard using the **Edit/Copy** menu choice, or send text that is already in the clipboard to the modem by choosing the **Edit/Paste** menu command.

Recording your Session

You may wish to keep a temporary or permanent record of your on-line connection. *QmodemPro for Windows* provides several ways to capture this data.

Capture file

The **Capture File** saves a record of all characters received to a disk file. As new information appears on the screen, that information is appended to the capture file. If you save your on-line sessions in a capture file, you can go back and refer to them later. This is a handy way to save bulletins, messages and other information to review after you've disconnected from the BBS or on-line service.

To open the Capture File, click on the **Capture** button on the tool bar above the terminal window. A Windows common dialog box will pop up, allowing you to assign a name, directory and drive letter to your capture file.

Scrollback

The **Scrollback** is very similar to the Capture File, except that your session is recorded in a memory (RAM) buffer instead of a disk file. You can view the Scrollback buffer and save copies on disk of the current screen, or the entire scrollback session.

To activate Scrollback, select **Terminal/Scrollback** or check the **Scrollback** button on the tool bar. When Scrollback is activated, a set of scroll bars appear on the right side of the screen, allowing you to look back over the recorded session.

There is a limit to how much data the scrollback buffer can store in memory during a session, and once that limit is reached, the oldest data is removed to make way for new incoming data.

Trap screen

If you need just a quick way to record a specific screen to disk, rather than recording an entire session in the Capture File, use the **Edit/Screen to File** command. An image of the screen will be saved in a disk file. A Windows common dialog box will pop up, allowing you to select the default file name or enter your own.

Log file

The **Log File** is a condensed report of your on-line session. When you set the **File/Log Toggle** to **on**, the log file records dates, times, names of files uploaded and downloaded, certain internal program events and error messages. A complete list of log file messages is in the Appendix at the end of this book.

Printing

The third way to capture data is to use the printer. Printing works in the same way as the Capture function, in that the entire on-line session is recorded, in this case to the printer.

Uploading and Downloading files

Uploading and Downloading refer to the act of copying a file from one computer to another over a modem connection, using a transfer protocol.

A transfer protocol is a set of signals and responses combined with the data being transferred. Protocols provide error checking and correction, and monitor the progress of the file transfer.

When you are sending a file from your computer to another computer, the process is called *Uploading*. A file is *Downloaded* when it is sent from another computer to your computer.

Batch transfers

Using the Windows file manager for batch uploads

Transfer protocols

Canceling a Transfer

Batch transfers

Some file transfer protocols allow you to transfer a batch of files during a single session. These protocols include Ymodem and Zmodem.

Although the exact procedure for starting a batch download on a remote computer varies from one system to another, it generally goes like this: First, select the files you want to download, by marking them or typing their names when prompted. If the remote system does not allow you to mark or tag files, go to the **Download** command on the remote system, select a batch transfer protocol such as Ymodem or Zmodem (if you haven't selected one already), then type the names of the files you want to download.

When your list of files is complete, you are ready to signal the remote computer to begin sending the files. Follow the instructions on the screen carefully.

Batch protocols can, of course, be used to transfer one file at a time. In all cases, you and the remote computer must use the same protocol, or the transfer will most likely fail.

Using the Windows file manager for batch uploads

An easy way to create a list of files for a batch upload is to use the Windows File Manager. The File Manager allows you to view the contents of drives and subdirectories, and when it is used with *QmodemPro for Windows*, you can drag and drop files from File Manager right into your Batch Upload window.

Try it now, just for practice. Launch *QmodemPro for Windows,* and the File Manager, and select **Upload** from the tool bar on the *QmodemPro for Windows* terminal screen.

Move your mouse pointer to the File Manager window, and select a file by pointing and clicking. Hold down the left mouse button while you drag the file to *QmodemPro for Windows*'s Upload window, then release the mouse button. Refer to your Windows User's Guide for ways to mark and drag multiple files from the File Manager. You may never have to type a file name again!

Note that the files in the *QmodemPro for Windows* file list are sorted alphabetically, and also note that you can't add the same file more than once to the upload list.

Transfer protocols

A number of file transfer protocols have been developed for personal computers, and *QmodemPro for Windows* incorporates some of the most popular, efficient, and reliable protocols.

Each transfer protocol is to some extent configurable, and a Setup option is provided for each one.

ASCII

Xmodem/Xmodem CRC/Xmodem-1K

Ymodem

Xmodem-1k/G & Ymodem/G

Zmodem

Kermit

CIS B+

Auto-start protocols

Some file transfer protocols, for instance Zmodem and CIS B+, are able to start receiving automatically without requiring you to start the download in *QmodemPro for Windows*. Other protocols such as Ymodem require you to begin receiving by selecting the **File/Download** menu command or clicking the **Download** button on the Tool Bar.

ASCII

The ASCII protocol is used primarily for on-line text transfers. It does not support transferring binary data such as programs or compressed files.

ASCII does not perform any error-checking, which makes it very sensitive to line noise. Bursts of phone line interference will introduce stray characters which are not part of the data being transmitted. For this reason, ASCII transfers are not recommended unless they are used for specific purposes where that protocol is needed.

It is advisable to use some type of flow control with ASCII transfers. Failure to do so may cause loss of data. Hardware **CTS/RTS flow control** is recommended.

Xmodem/Xmodem CRC/Xmodem-1K

There are three members of the Xmodem family of protocols in current use.

The original Xmodem uses a Checksum method to compare the received data with the transmitted data. Checksum is a very simple error-detection method with an accuracy rate of 99.6%.

Xmodem Checksum transmits 128-byte data blocks, and should only be used if the remote system offers no other choices.

Xmodem CRC (Cyclic Redundancy Checking) is similar to Xmodem-Checksum, including the 128-byte block size, but uses a far more reliable error-detecting algorithm (99.9969%). Like Xmodem Checksum, this is a relatively inefficient protocol, and should be used only if other options are not available.

Xmodem-1K is a modified version of Xmodem CRC, which uses 1024 byte blocks, as opposed to Xmodem's 128 byte blocks. You will sometimes see this protocol incorrectly labeled "Ymodem", with the true Ymodem protocol labeled "Ymodem Batch". This protocol is more efficient than the 128-byte versions of Xmodem, and is a good choice if true Ymodem and Zmodem are not offered on the remote system.

All three Xmodem protocols transmit only the contents of the file -- to receive (download) a file using one of these protocols, you must specify the name of the file both on the remote system, and on the local PC.

Ymodem

This protocol supports the transfer of multiple files within a single session which does not require user intervention.

Ymodem and Ymodem/G batch protocols include header records in the downloaded data. These records contain file information such as name, size, and date/time stamps, although you can control the date and time stamping of files from the **Options/Protocols** dialog box. As a result, file names need be specified only on the remote system before starting batch downloads.

Xmodem-1k/G & Ymodem/G

These two protocols are identical to their non-/G counterparts, with the exception that they do not perform any error correction, relying instead on the error-correction features of the modems. Modems equipped with internal error correction schemes such as MNP, ARQ, and LAP-M ensure error-free communications.

Xmodem-1K/G and Ymodem/G cannot re-send bad data blocks. In theory, there should be NO bad blocks of data in an error-free connection. In most cases, errors are caused by improper setup either in the hardware or in *QmodemPro for Windows*.

The /G protocols should not be used unless:

Your modem is capable of hardware error correction, such as MNP, ARQ or LAP-M

CTS/RTS flow control is turned **on** in the modem configuration.

Zmodem

This is the popular streaming protocol placed in the public domain by Telenet. Zmodem uses a variable sized block to transfer data over even the noisiest phone lines. Zmodem was designed as a Batch transfer protocol, and is able to send multiple files during a single session without operator intervention.

If an error is detected during a file transfer, Zmodem can tell the sending computer exactly where to restart the transmission. The real bonus to Zmodem is that it does not require a high-speed or error-correcting modem with MNP or V.42 to be effective. While it is not quite as fast as Ymodem/G or Xmodem-1K/G, it performs well enough to have become a standard in the industry.

One of the benefits of Zmodem is Crash Recovery. This allows an aborted file transfer to be restarted at a later time, without having to re-transmit the already received portion of the file.

It does not matter what protocol you used to begin the aborted transfer -- Zmodem will resume the transfer by comparing the partial file on disk with the new file being transmitted. If the two files are the same, Zmodem restarts the transfer where it left off. If the two files are different, Zmodem restarts the transfer from the beginning.

Kermit

This protocol's main claim is not speed, but rather its ability to communicate with many types of computers, from mainframes to microcomputers. It can cope with systems limited to seven-bit characters even when the data to be transmitted is in eight-bit form. All characters to be sent are translated into standard printable characters and reconstructed on the receiving end.

While not terribly efficient, Kermit is sometimes an absolute necessity for data transfers involving different types of systems and terminal types. It is not normally recommended for PC to PC transfers because it is much slower than other protocols.

CIS B+

CIS B+ is a high-efficiency error correcting transfer protocol available on CompuServe. It uses the very reliable Cyclical Redundancy Check (CRC) mode of error correction for binary and text file transfers. This protocol is also able to resume an aborted download, sending only the number of bytes necessary to complete the file transfer.

CIS B+ transfers are controlled by the host -- you will be asked to enter the complete path and filename for the file on your system before the transfer begins. This applies to both uploads and downloads.

Canceling a Transfer

You can cancel (abort) a file transfer by clicking the **Cancel** button on the Upload or Download Status Dialog Box.

Canceled uploads are handled smoothly -- the host system will recognize within a few seconds that the transfer has failed, and you be back where you were before you began the upload.

If you cancel a download, the host system may not immediately recognize the transfer failure, and may continue to send binary text to the screen. Hold down the **[Ctrl]** key and tap the **[X]** key several times to signal the host to cancel the transfer and return to the normal prompt.

How do I send a file to my buddy?

It's really quite easy to send files back and forth between computers. You'll need to arrange beforehand which computer will do the calling and which will do the answering. Both computers need to be running communication software -- you, of course, will be running *QmodemPro for Windows.*

And you'll both need to agree on what **Transfer Protocol** you're going to use for the session. We recommend **Zmodem**, if your friend's communication software offers this protocol, if not, then Ymodem (batch) or Xmodem-1K would be good choices. Just be sure both of you are using the same transfer protocol.

Your friend's communication software must be capable of going into **Terminal Mode**. In other words, it must allow the user to type commands to the modem, and see the modem's response. Any version of *Qmodem* or *QmodemPro* will work just fine, and many other communication programs will do the job also.

For the purposes of this example, we'll assume you are going to be dialing your friend's computer. Simply reverse the procedures below to answer a call or receive a file.

Dialing

Dial your friend's computer after arranging with him to load his communication software and go into Terminal Mode. You can dial either by creating a phonebook entry and dialing from there, or you can use the **Phone/Manual Dial** function.

Answering the phone

Your friend's modem will detect the incoming RING signal, and display the word RING on screen. The modem may or may not answer the call itself, depending on whether it is in Auto Answer mode. If it doesn't answer, ask your friend to type

`ATA` **[Enter]**

at the terminal screen. This command simply tells your friend's modem to answer, and negotiate a connection with your modem.

Both modems will beep and squawk as they **handshake** and establish a connection. When both modems have responded with something like

`CONNECT 19200/REL`

or

`CONNECT 2400`

or whatever speed and error correction features they have in common, you're ready to begin sending the file.

How to send the file

If you're sending a file to your friend's computer, start the upload by clicking the **Upload**button on the tool bar. Select the protocol, then select the file or files you are going to send. When you are satisfied with your selection, click the **Send List** button.

How to signal your friend's computer to receive the file

If you are both using Zmodem, in most cases the file transfer will start automatically on your friend's computer. If it doesn't, your friend will need to start a **download**, or **receive** the file. In most DOS communication programs, the **[PgUp]** key or **[Alt][R]** will start a download. When your friend has selected the appropriate transfer protocol and confirmed his selection, the file transfer will begin.

How to disconnect

Use the **<u>Terminal/Hangup</u>** button when you have finished sending files. That's all there is to it!

Viewing a file

QmodemPro for Windows allows you to view text and graphics files on your disk in a pop up window. The text files you might want to view include the capture file, log file, scripts, and note files.

Text files (binary files too)

QmodemPro for Windows has a simple file viewer, launched from the **File/View a File** menu command, which allows you to look at text or binary files. From the viewer menu, you can search for text within the current file, and you can mark blocks of text and copy them to the clipboard to paste into other screens or applications. You have the option to view files in text mode or in hexadecimal mode -- this may be useful when you are looking at the contents of a binary file.

To open another file from the File Viewer, select the **File/Open** menu, or drag a file icon from the Windows File Manager into the File Viewer window.

Graphic files

Many Bulletin Board Systems and on-line services have file sections devoted to graphic files. These files contain digitized reproductions of drawings, photographs, or even computer art.

Many computer users enjoy downloading and looking at picture files. *QmodemPro for Windows* helps you do this by automatically displaying certain types of graphic files in a window on your screen as the file is being downloaded.

If you want to examine a part of the picture in more detail, you can zoom in by clicking the **Zoom** button on the tool bar. Move your mouse pointer to the area you wish to enlarge, then click the left mouse button. Clicking the left mouse button enlarges the part of the image under the mouse pointer, while clicking the right mouse button shrinks the image.

You can use the GIF viewer to look at graphic files already on your disk, as well. Select the **File** menu option **View a GIF file** to pop up the viewer window. From there, you can select one or more GIF files to view, either one at a time, or multiple files.

To open another file or group of files from the GIF Viewer, select the **File/Open** menu, or drag one or more file icons from the Windows File Manager into the GIF Viewer window.

Part or all of an image can be copied to the clipboard for use in other applications. To grab a portion of an image, position the mouse pointer over the top left corner of the area to mark, and hold down the left mouse button as you drag it to the bottom right corner. Select **Copy** from the tool bar or edit menu to put the selected image on the Windows clipboard. Select **Copy** without marking an area to grab the entire image.

What to do with compressed files

Most on-line services and Bulletin Board Systems compress program and document files in their library to save space on disk and reduce transfer time. These file compression programs analyze the contents of a file, and replace repetitive patterns of characters with simpler patterns, thus making the file smaller. Another function of file compression is to create an envelope containing more than one file inside it.

Since most programs or applications actually consist of several files, this is the most reliable way to ensure that when you download a program or application, you get the complete program, on-line documentation, manuals, information files, and so on.

But before you can actually install and run the program, you will need to un-compress the file. That means you'll need a copy of the same program used to create the compressed file in the first place. Most of these compression programs are shareware or freeware, and can be downloaded from your favorite local BBS or on-line service.

One of the most popular file compression utilities is PKZIP from PKWARE. You can always download the latest version (currently PKZ204G.EXE), from MSI HQ BBS and most local Bulletin Board Systems. To use this file, copy it to a directory in your DOS path, and run it in that directory by typing

> `PKZ204G` **[Enter]**

This file is a self-extracting archive containing the program, utility, and documentation files you need to use PKZIP and PKUNZIP. Most other compression utilities work in the same way.

If you need help understanding the use of DOS commands such as COPY and PATH, refer to your DOS manual.

Quick Learning Scripts

The **QuickLearn** Script Recorder allows *QmodemPro for Windows* to learn how to log onto a BBS or On-line Service for you. It does this by recording your response to data that comes from the host system, and writing the incoming data and your responses as a simple script, or program.

The script language itself is very much like Basic, so those already familiar with this language will find scripting easy. For those users interested in learning more about the script language so they can write their own scripts and perform more complex tasks, we encourage you to review the **SLIQ Script Language Reference.**.

There are two ways to begin QuickLearning a script. The first is to go to the **Scripts** menu and select **QuickLearn.** The second way is to go to the Phonebook, and **Revise** the entry you want to dial. Enter a filename in the Scripts field in the Phonebook entry. The next time you dial this entry, *QmodemPro for Windows* will automatically record a script file.

Many QuickLearn scripts are ready to run without further editing or modification. If your newly-learned script requires editing or debugging, please refer to the SLIQ Script Language topic, or volume 2 of the documentation.

Opening a DOS Window

QmodemPro for Windows allows you to open a DOS window right from the File menu. Select **File** and then select **<u>Shell to DOS</u>**. *QmodemPro for Windows* will execute the Windows command DOSPRMT.PIF and open a DOS window for you.

Sending and Receiving Fax Messages

QmodemPro for Windows is able to send and receive Fax messages if your modem hardware supports CLASS1 or CLASS2 Fax.

Quick Fax

Sending a Fax from the Phonebook

Receiving a Fax

Viewing a Fax

Printing a Fax

Quick Fax

You can send a fax quickly using the **Fax/Send Fax** menu command. A dialog box will pop up asking you for the Fax number to dial, and information such as the sender and recipients' names for the cover sheet, and the name of the document to send. When you finish entering the information, click the **OK** button, and *QmodemPro for Windows* will begin dialing the Fax number you supplied.

QmodemPro for Windows will then convert the characters and graphics in the file to a format which can be sent as data over a modem connection and translated into a graphical representation of your original document on the receiver's Fax machine. This process is called *rasterizing.*

When your modem successfully connects with the Fax machine at the remote end, it will display a dialog box showing the connection speed and the progress of the Fax transmission. When the Fax is finished sending, *QmodemPro for Windows* will disconnect automatically.

Sending a Fax from the Phonebook

If you want send Fax messages to the same phone number on a regular basis, it's a good idea to put that information in the Phonebook. Create a phonebook entry for the Fax machine you wish to call.

The **Fax Device Only** selection must be toggled **on**, so that *QmodemPro for Windows* knows to enable the Fax portion of your modem for this call. This is the case even if you are calling another Fax modem.

When you dial a phonebook entry tagged **Fax**, *QmodemPro for Windows* will pop up a dialog box asking for the name of the document to send. The document must consist of plain ASCII text, or it may be a PCX graphics file or a BMP Windows Bitmap file.

Fill in the names of the sender and recipient, and the subject of the message. The To, From and Subject fields are automatically imported into the cover sheet.

The option box for **Fine Mode** sends the Fax at a higher resolution, allowing a sharper image to be transmitted, but this increases the time it takes to send the document.

Check the option box for **Include Cover Page** if you want to send a cover sheet with your Fax.

Receiving a Fax

There are two ways to receive Faxes: manually, and automatically. To receive Faxes automatically, your current device must be configured for Fax. Go to the **Fax** menu on the Terminal window, and select **Auto-Receive**. Then whenever your phone rings, the modem will answer, negotiate a Fax connection, and receive the Fax.

You may only want to receive Faxes by arrangement with the person sending the Fax, rather than from anyone at any time. The **Fax/Receive** menu command will answer the phone, receive the incoming Fax, and then exit back to the Terminal window.

In both cases, incoming Faxes are stored as files on your disk. The files are placed in your **Fax Receive Path**, which you specify from the **Options/Fax** menu. To see a list of received Faxes, go to the **Fax** menu and select **Receive Log**. This dialog box lists each Fax, the date and time it was received, the number of pages received, and the sender's ID (usually the phone number).

Select Faxes to view or print by clicking the left mouse button over a file in this list.

Viewing a Fax

The <u>Fax viewer</u> can display your Faxes in a number of ways. The default mode is to show each page full size on your screen. You can reduce the image to 25% or 50% of its original size, or enlarge it to full size, or display a thumbnail image of each page of a multi-part Fax message. You can even copy the current page to the Windows Clipboard to paste into another application. Each of these commands has a corresponding button on the Tool Bar.

Printing a Fax

You can print a Fax either from the <u>Fax viewer</u>, or directly from the **Receive Log** or **Auto-Receive** dialog boxes. Select the **File/Print** command from the Fax viewer, or the **Print** button from the Receive Log.

The Print command uses the **Print Dialog Box**. It allows you to define resolution, a range of pages to print, and whether to print multiple copies of the document. A **Setup** button pops up a dialog box with setup information for your currently selected printer. From there, you can select a different printer, or change the printer's internal settings to print the file the way you want.

Devices

A Device is simply a set of modem configuration instructions for a particular modem, attached to a particular port on your PC. You may define more than one device for a single modem, and you may define one or more devices for multiple modems on your PC.

Rather than having to specify all the necessary communication parameters manually, or separately for each Phonebook entry, these parameters are associated with a Device. Devices are then associated with Phonebook entries.

Anyone who has ever bought a new modem, or moved their modem to different serial port, and had to re-enter all that configuration information into all their phonebook entries, will appreciate this modular approach. Simply change the necessary parameters associated with a device (from the **Options** menu, select **<u>Device</u>** and edit your settings), save your changes, and the work is done!

<u>Pre-defined devices</u>

<u>Cloning a device</u>

<u>Alternate Dialing Prefixes</u>

Pre-defined devices

QmodemPro for Windows has two predefined devices. The first of these is **-Default-**. If you select this as the **Device to use** when you add or revise a <u>Phonebook</u> entry, you can easily copy your Phonebook to another computer, or use a different modem, without reconfiguring your phonebook entries.

The second is **-Current Device-**. This uses whatever device is currently selected from the **<u>Device/Select</u>** menu.

Cloning a device

In some cases, the settings in the default device aren't right for a particular host system. For instance, if most of your calling is to Bulletin Board Systems, you've probably set your default device to 8 data bits, no parity and 1 stop bit. If you want to call a mainframe computer or an on-line service, you will probably need to use 7 data bits, even parity and 1 stop bit.

The simplest way to do this is to create a copy of your default device with these special settings. Go to the **Options** menu, select **Device**, and select your default device from the list. Click on the **Duplicate** button to create a copy of the device.

Give the device a new name which describes its purpose (for instance 19200 7E1). Then select the **Hardware** button and select your new parameters. In this case, we'll be changing the parity to Even and data bits to 7. Save your settings by clicking the **OK** button, and return to the Phonebook by clicking the **Close** button .

Now, it's a simple procedure to select and revise your phonebook entries and attach the newly-cloned device to those entries which use 7-E-1 instead of the default 8-N-1.

Alternate Dialing Prefixes

QmodemPro for Windows allows up to 10 Prefix codes per device, assigned to the letters A through J. Each prefix code can store up to 30 characters. To edit dialing prefixes, go to the **Options** menu, select **Device**, and select the device you want to edit from the list. Click on the **Prefixes** button to edit the dialing prefixes.

You can use these prefixes to send long distance access code numbers, PBX codes, extension numbers, and more. Note that these prefixes can be placed anywhere in the phone number. For instance, if you dial 9 from your PBX to get an outgoing phone line, you may wish to add "9," (dial 9, then wait two seconds) to your list of prefixes.

If you need to enter a phone credit card number after dialing a number, you can store that card number as a prefix (although in this case it is actually used as a suffix), then append the prefix letter to the phone number, to save having to type the card number into every phonebook entry.

Do *not* define a phone number using its alpha representation. A number such as "1800DECDEMO" may be a valid telephone number, but the dialer will substitute the prefix values of letters A through J in any phone number.

Special control characters may also be entered in these boxes, if necessary. See **Special Control Characters** for more information on how to enter control code sequences in macros, modem command strings, and scripts.

You will notice in the previous illustration that the Prefix codes C, D, and E, appear to be modem dialing commands. These commands can be used to override the default dial command for the device specified for a phonebook entry.

To use these alternate dialing commands, precede the desired prefix in the phone number entry field with the special character ` (reverse apostrophe or single open quote). When *QmodemPro for Windows* detects that special character it will use the prefix code immediately following instead of the default dialing command.

The example above assumes a modem with an MNP error-correction mode that is enabled with the AT&M4 command, and is disabled with the AT&M0 command. The DT "Dial Tone" command tells the modem to tone-dial the number that follows.

In the example above, the primary number and alternate numbers 1 and 2 will be dialed with an AT&M4DT command, which enables MNP error-correction and dials the Mustang Software HQ BBS (which operates a few HST modems). Alternate #3 will be dialed with an AT&M0DT command, which disables MNP error-correction and dials 1-805-873-2403

If you use the reverse apostrophe modifier to enable or disable one or more of your modem's features, be sure to set the default dialing prefix to the one you use most often. By doing this, you only need to override entries that do not conform to the default mode. In the previous example, your default dialing prefix should be "AT&M4DT". Prefix `D would be used to dial non-MNP systems.

Fine tuning

The default settings for *QmodemPro for Windows* are designed to accommodate the majority of users, but most areas of the program are completely configurable, including the terminal parameters, device, and desktop.

Why change the default settings?

Because some systems you call may have special requirements not provided for in the default settings. You may, for instance, be calling a mainframe computer or on-line service which does not support 8 bit connections.

Or, you may be required to use a special Terminal Emulation to run applications on another computer, with specially redefined function key commands and display attributes.

Finally, you may, for your own convenience, want to assign certain frequently-used commands or text to a macro key, for instant recall.

How do I know if I need to make changes?

The following paragraphs explain the meaning of each of the parameters you can change in the Terminal Window, along with examples showing why you might need to change these parameters.

Changing port settings

Other Device Settings

Changing terminal settings

Emulations

Changing port settings

The **Device/Fine Tune** menu allows you to make changes on the fly when necessary, and overrides the current device settings.

Baud rate

The terms "Baud" and "Bits per Second" are used interchangeably, though this is not necessarily accurate usage. Baud is short for "Baudot", a term which describes the number of discrete signal events per second occurring on a communications channel.

Each signal event generally transmits one Bit of information. Most higher speed modem definition standards allow for more bits to be transmitted for each signal event, which means that **Baud** and **Bits Per Second** are not always the same.

In all cases within *QmodemPro for Windows*, the Speed refers to Bits per Second, abbreviated as "BPS". For instance, your modem may be capable of 2400 BPS, or 14400 BPS, or 9600 BPS.

The baud rate is the speed at which the COM port operates and should in most cases match the top speed of the device, or the locked DTE speed for high speed modems. If DTE is not locked, *QmodemPro for Windows* will automatically adjust to a lower speed upon connection, if necessary. Supported values are 110, 300, 1200, 2400, 4800. 9600, 19200, 38400, 57600, 115200 and 230400, although unless you install high performance communication drivers to replace the Windows communication drivers, we recommend setting the Baud Rate no higher than 19200.

High speed modems, that is, modems with nominal baud rates of 9600, 14400, 16800 and higher, should always be set to a higher baud rate than the actual connect speed, to enable hardware flow control, data compression and error correction. Thus, the correct setting for a 9600 or 14400 baud modem would be 19200 or higher.

Parity

Parity is used to insure the validity of the data being transmitted and received. The Parity is calculated by performing a simple arithmetical computation on the bits received, and comparing the result with the Parity Bit sent with each group of Data Bits. Parity values are None, Odd, Even, Mark and Space.

Most Bulletin Board Systems use None (no parity), and both systems must match. A parity mismatch will cause your screens to appear garbled.

Stop Bits

Stop bits are generated by the serial port chip or UART (Universal Asynchronous Receiver Transmitter) to isolate transmitted characters. In other words, since each character is composed of a number of bits (usually 7 or 8) transmitted one at a time, the programs need stop bits to tell the sending and receiving computers where each character begins and ends in the data stream.

Data bits

Characters transmitted over phone lines are made up of **data bits**. Most PC-based Bulletin Board Systems use 8 data bits, while some mainframe and minicomputer systems use 7 data bits. Both systems must match -- if one side is expecting 8 data bits

and the other side is expecting 7, your screens will be garbled.

Other Device Settings

Break signal

"BREAK" is a special signal generated by the serial port, used to signal the remote computer to have it handle some special condition. In most cases, **BREAK** is used to tell the other computer to interrupt whatever it is doing. Most PCs do not recognize the **BREAK** signal, although a few older mini and mainframe systems still do.

As is the case with PC programs which ignore or disable the **[Ctrl][Break]**, the BREAK signal may not be recognized by the receiving computer. BREAK handling is a function of the communication software on the remote system, and this feature may not be implemented on the computer you are communicating with.

Flow Control

A real-life example of flow control might be a conversation on a CB Radio channel, where each party indicates he has finished speaking by saying "Over". In this way, the message comes through without interference.

Communication devices must also signal each other when they are ready to send and receive data, so that both systems remain synchronized and no data is lost.

XON/XOFF are software controlled transmit/receive signals which control the flow of text or data. While **XON/XOFF Flow Control** is active, **[Ctrl][S]** will stop the flow of data and **[Ctrl][Q]** will restart it. Note that software flow control is not used by most communication software.

XON/XOFF is generally used to control text viewed on the screen or to prevent a data buffer overflow, and is controlled in **Device/XonXoff** and **Options/Device/Hardware**.

Full/Half Duplex

Duplex describes the method used to echo your keystrokes back to your terminal screen. In **Full Duplex** mode, the remote system echoes (displays) your keystrokes back to you. In **Half Duplex** mode, the remote computer does not echo your keystrokes, so *QmodemPro for Windows* must do it instead.

An incorrect Duplex setting will yield one of two results: either you will NOT see any keystrokes displayed on your screen as you type, or your keystrokes will be duplicated, like this:

```
HHEELLLLOO
```

In the first case, you are operating in FULL duplex mode, and you should switch to HALF. In the second case, you are operating in HALF duplex mode and you should switch to FULL.

Just to make this more interesting, most Hayes-compatible modems also feature an internal Echo setting. If the modem's echo is set to **on**, the modem itself will echo your keystrokes back while in command mode. If your *QmodemPro for Windows* settings are correct, but your keystrokes are still duplicated, the modem is likely to be the culprit. Check your modem manual for switch settings or initialization parameters to turn the local echo **off**.

Changing terminal settings

Doorway mode

Doorway mode is a special emulation that allows your PC to send keyboard scan codes such as function keys and cursor movements to the host system, rather than having those keys interpreted as commands in the Terminal window. To operate properly, the host system (the system you are calling) must have support for the doorway emulation as well.

This allows you to use the full screen editing keys in many Bulletin Board System message editors, and to run certain programs by remote control.

Carriage Return/Line Feed handling

Different kinds of computers use different methods for indicating the end of a line of text. On IBM compatible PCs, two end-of-line characters are required: the carriage return (13 decimal) and the line feed (10 decimal). Some other kinds of computers, for instance some mini and mainframe systems and the Apple Macintosh, use only a carriage return with no line feed character.

The result, if line feeds are not being sent, is that text on your screen will be overwritten. By turning **on** the **Add LF to CR** toggle, *QmodemPro for Windows* will automatically add a linefeed character when it receives a carriage return.

The **Options/Emulations** menu offers additional line feed selections based on the selected terminal emulation, and the **Options/Protocols/ASCII Options** dialog box allows total customization of carriage return and line feed handling during ASCII data transfers.

8th Bit Strip

Computers ordinarily use one byte, or 8 bits to display a single character on the screen. IBM compatible PCs use all 8 bits to indicate a single character, while many mainframe systems use only 7 bits for a character. The 8th bit is also called the "high bit".

PC compatibles use the 8th bit to add another 128 characters to the ASCII character set. These characters are called "high ASCII" or "IBM ASCII" characters, and are used to provide accented characters, mathematical symbols, and line and box drawing characters.

Mainframe computers, on the other hand, generally only use the first 7 bits, and either ignore the 8th bit, or use it for some other purpose, such as parity checking. This may cause your screen to appear cluttered with foreign language or graphic characters. Turn **on** the **8th bit strip** in this case, to display normal ASCII characters.

Emulations

The **Terminal/Emulation** feature allows *QmodemPro for Windows* to behave like a terminal other than a standard PC display, such as the popular VT100 from Digital Equipment Corp. While in terminal emulation mode, *QmodemPro for Windows* interprets keyboard and video information so that is displayed properly on the PC screen, as if it were a native terminal of the selected type.

QmodemPro for Windows supports more than 30 different types of emulations. For most purposes, you should be using ANSI, unless you are sure another emulation is required. Use Vidtext or ANSI when you call CompuServe.

Another emulation you may find useful on Bulletin Board Systems that support it is RIPscrip. This is a graphical emulation with full mouse support. RIPscrip is available in most quality BBS software programs, and when properly configured, it allows users to select menu options with the mouse, and see full-screen graphics.

If you are connecting with a mainframe computer, you may need to select some other terminal emulation. Your system administrator or PC Support personnel can advise you on the recommended terminal emulation for the mini or mainframe system you will be calling.

The following emulations are supported by *QmodemPro for Windows:* **ADDS VP60, ADM3A, ANSI, Avatar, Debug ASCII, Debug Hex, DG 100, DG 200, DG 210, Hazeltine 1500, Heath 19, IBM 3101, RIPscrip, TTY, TVI 910, TVI 912, TVI 922, TVI922, TVI 925, TVI 950, TVI 955, Vidtex, VT 52, VT 100, VT 102, VT 220, VT 320, Wyse 30, Wyse 50, Wyse 60, Wyse 75, Wyse 85, Wyse 100,** and **Wyse 185**

Customizing the desktop

Here, we'll show you some of the changes you can make using the **Options/Desktop** menu choice..

Patterns and Colors

The desktop pattern is a simple repeating geometric pattern made of black and white squares. You can select from a number of different patterns in the list box at the top corner of this dialog box by clicking on the arrow at the right side of the box.

Select **None** if you don't want to have a background pattern, or select one of the patterns in the list, if you prefer.

The **Colors** option allows you to change the colors that make up the pattern. Use the list boxes to select foreground and background colors.

Wallpaper

The **Wallpaper** option allows you to select a Windows .BMP file to display on the Desktop. The bitmap image is displayed on the desktop surrounding the terminal screen. For a repetitive pattern made up of a small bitmap, select **Tile.** To center a larger bitmap on your desktop, select **Center.** If wallpaper is used, the selection of patterns and colors may be covered in part or entirely by the wallpaper.

Customizing Sounds

If you have a sound card in your PC, *QmodemPro for Windows* can use it to play WAV files to signify system events such as a successful connection, the completion of a file transfer, and more.

Go to the **Options** menu and select **Sounds** to make changes. A dialog box will pop up showing a list of sound events you can modify. Highlight the entry you want to edit, and click the **Change** button. A standard Windows Common Dialog Box will pop up showing a list of WAV files in the left window, and options to search different drives and directories in the right window.

A **Test** button allows you to listen to the selected WAV file before committing yourself to each selection.

Using a network modem with QmodemPro for Windows

QmodemPro for Windows can also be used on modems connected to your Local Area Network. In some cases this may require special software drivers on both the network and/or your workstation. This allows you to use modems on the LAN as if they were connected to your own PC.

Since each network installation is different, a discussion of specific network modem drivers and operational parameters is beyond the scope of this manual. If you are on a Local Area Network, and would like to use *QmodemPro for Windows* on a network modem, ask your Network Administrator for information on what drivers are required, and how to install them on your PC.

Support for Interrupt 14 is included in *QmodemPro for Windows* and does not require any special software.

Command line options

You can start *QmodemPro for Windows* with command line options that specify phonebook files to load, entries to dial, and scripts to execute. In this way, you can create an icon for your *QmodemPro for Windows* program group that automatically executes the Host script, for instance, or dials a list of phonebook entries, or loads a different phonebook file.

Any scripts attached to individual phonebook entries will execute as *QmodemPro for Windows* connects successfully with each entry.

Dialing a single Phonebook entry

The simplest command line switch simply starts up *QmodemPro for Windows,* specifies the phonebook file to load, and dials a specific entry. Entries are specified by number. For example, to dial entry number 4 in the default Phonebook file, the command line looks like this:

```
QMWIN QMWIN.PHN 4
```

Loading a different Phonebook file

The default Phonebook file is called QMWIN.PHN, and this phonebook is loaded automatically when you start *QmodemPro for Windows*, unless you specify otherwise.

If you have created other Phonebook files, you can load one of these in place of the default QMWIN.PHN from the command line.

Say you have a phonebook file named LOCAL.PHN containing only local phone numbers. To start up *QmodemPro for Windows* with your LOCAL.PHN Phonebook loaded, use a command line like this:

```
QMWIN LOCAL.PHN
```

Dialing multiple entries

There are two ways to auto-dial a group of Phonebook entries -- by loading and dialing a group of entries, or by searching for text in the Phonebook, then tagging and dialing entries which contain the search text.

By Group name

Now, say you want a command line that will tag and dial every system in the group named "Cool Files". You would call your group from the command line like this:

```
QMWIN QMWIN.GRP "Cool Files"
```

This tells *QmodemPro for Windows* to load the Group file associated with the default Phonebook, then tag and dial all members of the group "Cool Files".

You can, of course, use a group file associated with a different phonebook, by typing the phonebook name with the extension .GRP in place of the default QMWIN.GRP. A command line loading the LOCAL phonebook and dialing all members of the group "Great Games" would look like this:

```
QMWIN LOCAL.GRP "Great Games"
```

For a refresher on how to create and edit Groups, refer to the Phonebook section of the

previous chapter.

By Text Search

This works in the same way as the **Search** menu command in the Phonebook. Load the phonebook from the command line, together with the text you want to find, in quote marks.

For instance, if you want to dial everything in the default QMWIN.PHN Phonebook matching the text "new BBS", use a command line like this:

```
QMWIN QMWIN.PHN "new BBS"
```

As with previous examples, you can load a different phonebook in place of the default QMWIN.PHN.

Running a Script

To execute a script automatically when you start *QmodemPro for Windows*, simply specify the name of the script on the command line, like this:

```
QMWIN SCRIPT.SCR
```

or

```
QMWIN C:\QMWIN\SCRIPTS\SCRIPT.SCR
```

Note that unless a full path is included, the script file must exist in the default script directory, specified in the **Options/Files/Path Definitions** dialog box, and the extension .SCR or .SCC is required.

Please refer to Volume 2 of this documentation for more information on the *QmodemPro for Windows* script language.

Creating an Icon

The final step in automating your *QmodemPro for Windows* sessions is to create icons which start *QmodemPro for Windows* with command line switches. Here's how to do it:

From Windows, go to the **Program Manager**, and select a **Program Group**. Then click on the **File** menu, and select **New.** Then select **Program Item**, and type in the **Description** you'd like to assign to your new task.

In the text box for **Command line,** type

```
QMWIN
```

followed by the appropriate command line argument.

For **Working Directory,** type the drive and path in which your *QmodemPro for Windows* application is installed. The **Shortcut key** is optional, but allows you to execute the icon from a keystroke such as **[Ctrl][Alt][Q]**.

Then select **Change Icon,** and type in the path and filename of the icon you'd like to assign to this task. You can use the **Browse** button to see a list of files containing Windows icons. The file BBSICONS.DLL supplied with *QmodemPro for Windows* contains several dozen BBS icons for you to use if you like. When you are satisfied with your selection, click **OK** to save your new icon.

To execute your customized *QmodemPro for Windows* startup command, simply double-click the new icon from the Program Manager.

The Editor

QmodemPro for Windows has a simple built-in text editor, suitable for writing and editing scripts, batch files and other ASCII text files.

Keyboard commands

If you have used other word processors and text editors, you will probably already be familiar with how to use the arrow keys, **[PgUp]** and **[PgDn]** keys to move the cursor around the screen. Here are some additional keyboard commands you may find useful.

[Ctrl][N]	Split line
[Ctrl][T]	Delete word right
[Ctrl][V]	Paste from clipboard
[Ctrl][X]	Cut marked text to clipboard
[Ctrl][Y]	Delete line
[Ctrl][Del]	Delete block
[Shift][Del]	Cut to clipboard
[Shift][Ins]	Paste from clipboard
[Ctrl][Ins]	Copy to clipboard
[Alt][Bkspace]	Undo editing
[Shift][Alt][Bkspace]	Undo undo
[F3]	Find text
[Ctrl][PgUp]	Top of screen
[Ctrl][PgDn]	Bottom of screen
[Ctrl][Home]	Beginning of file
[Ctrl][End]	End of file

Marking Text

Move the insertion point to the beginning of the text you want to mark using arrow keys or mouse.

To mark text using the mouse, hold down the left mouse button while you drag the mouse to the end of the text you want to mark. Marked text will show up in a contrasting color. When you have finished marking text, release the left mouse button.

To mark text using the keyboard, hold down the **[Shift]** key while you move the insertion point using the arrow keys to the end of the text you want to mark. When you are finished marking text, release the **[Shift]** key. **[Shift][Home]** marks everything from the current

cursor position to the beginning of the line, while **[Shift][End]** marks everything from the current cursor position to the end of the line. **[Shift][Ctrl][Home]** marks everything from the current cursor position to the beginning of the file, while **[Shift][Ctrl][End]** marks everything from the current cursor position to the end of the line.

Moving Text

First, mark the text you want to move. Then cut the marked text to the clipboard using your mouse to select the menu command **Edit/Cut**. Or, press **[Shift][Del]**.

Then move the insertion point to the place where you would like to move the selected text. Using your mouse, select the menu command **Edit/Paste**. Or, press **[Ctrl][Ins]**.

Copying Text

First, mark the text you want to copy. Then copy the marked text to the clipboard using your mouse to select the menu command **Edit/Copy**. Or, press **[Shift][Ins]**.

Then move the insertion point to the place where you would like to copy the selected text. Using your mouse, select the menu command **Edit/Paste**. Or, press **[Ctrl][Ins]**.

Deleting Text

Use the **[Bkspace]** or **[Del]** key when you want to delete text from the current document but you have text on the Clipboard that you want to keep.

To delete one character at a time

Press **[Bkspace]** to delete the character to the left of the insertion point. Press **[Del]** to delete the character to the right of the insertion point.

To delete more than one character

Select the text you want to delete. Press **[Bkspace]** or **[Del]**. To place the text onto the Clipboard, choose **Cut** from the **Edit** menu.

Undoing changes

You can **undo** changes, and restore your text to the way it was before you performed the most recent command, with the **Edit/Undo** command or **[Alt][Bkspace]**.

You can undo multiple levels of previous commands, starting from the most recent and working backwards. The number of times undo can be used depends on the amount of information affected by the commands. Up to 8K of previous information is stored in the undo buffer.

Formatting Text

To format text, do the following.

End a line   Press **[Enter]**.

Indent a line   Press **[Tab].**

Insert a tab   Press **[Tab].**

Insert a blank line then press **[Enter]**.	Move the insertion point to the beginning or end of a line, and

Finding Text

You can start a search for specific text at any point in a document.

To find specific characters or words, move the insertion point to where you want the search to begin. From the **Search** menu, choose **Find**. Type the characters or words you want to find. If you want to match capitalization exactly, select the **Case Sensitive** check box.

To specify the search direction, select the **Forward** or **Backward** option. Then choose the **Find Next** button.

To find the next occurrence of the text, choose the **Find Next** button again. You can also close the Find dialog box and choose **Search Again** from the **Search** menu or select the toolbar button **Repeat**.

Changing the appearance of the terminal screen

The size of the actual terminal screen on the *QmodemPro for Windows* desktop depends on the font you have selected for the current terminal emulation, along with the number of lines and columns specified. Usually, this is 80 columns wide by 25 lines deep.

The terminal screen always uses monospaced fonts rather than proportional fonts. In other words each character in the terminal screen has the same width, like on an old typewriter -- as opposed to proportional fonts used in typeset documents such as the screen you are reading right now, in which characters may vary in width from a lower case "i" to a capital "W".

Thus, the boundaries of the terminal screen are indicated by a border, and the screen is centered in the *QmodemPro for Windows* desktop. You can enlarge and reduce the desktop surrounding the terminal screen by dragging a corner of the window with your mouse. As you do so, the font size changes to accommodate your new window size.

If you prefer to see only the terminal screen without the surrounding desktop, you can resize the window to the exact size of the terminal screen borders by clicking the arrow at the bottom right corner of the window. This arrow appears only when the window is *smaller* than terminal screen, and has scroll bars on the bottom and right edges of the window.

Changing the font

The size of the terminal window is governed by two things: the font selected to display the characters on the screen, and the number of rows and columns you define for your terminal window. To change the font, go to the **Options** menu, and select **Emulations.** Select the emulation you would like to edit, and click the **Font** button.

Note that only monospaced fonts will be available for selection. An example of the selected font is displayed in a window in the bottom of the dialog box. Larger fonts will enlarge the area of the terminal window on your screen.

Select the font and size you prefer, click the **OK** button, and then view the results of your changes on the terminal window. Note that by selecting **Options/Desktop/Terminal Font/Based on window** the size of the selected font can be changed by enlarging the terminal window -- grab an edge and move it with the mouse to change the size.

Special RIPscrip fonts

RIPscrip terminal emulation mode has some additional features you might like to explore. On the **Options/Emulations** screen, an additional **Options** button appears on the right side of the dialog box when you select **RIPscrip** emulation.

Select the **TrueType fonts** option to use high-quality scalable fonts in place of the lower resolution RIPscrip fonts.

Caution: The RIPScrip emulation is not designed to be used with TrueType fonts. If you elect to use the TrueType fonts as replacements for the standard RIPscrip fonts you may experience screen display problems. These problems come about since the size and shape of the TrueType substitute fonts do not match the RIP fonts pixel for pixel.

The types of problems you may experience includes slightly offset text placement and overlap of text when multiple fonts are used. In cases where the screen uses an open-

face font that has been filled with a color, you will probably find that the entire screen fills with the color. This is because the point selected for filling is no longer located within the boundaries of the original font character.

If you find these problems are happening regularly on the systems you call you should make use of the real RIPscrip font characters rather than the TrueType substitutes. The RIP emulation is not designed to use TrueType fonts, and our option to do so is an effort to offer an alternative higher resolution.

Changing colors

You can change the attributes of the foreground and background text colors in the terminal window, by selecting **Options/Emulations**, selecting the emulation you would like to edit, then click the **Color** button.

The foreground and background lists show all the colors available, and the sample text window allows you to preview your choices.

Keep in mind that the remote host is capable of changing the screen colors during an on-line session, and will override your local settings.

Changing the rows and columns

To change the number of rows and columns displayed in the terminal window, go to the **Options** menu, and select **Emulations.** Select the emulation you would like to edit, and make your changes in the **Terminal Size** box in the dialog box. The screen width can be changed from the default 80 columns to 132 columns, but this is useful only if the host system you call supports a screen width of 132 columns. You can also change the number of lines displayed on the screen from the default of 25. When you do this, be sure to change your user settings on the host system to accommodate the new screen length.

There are several other ways to modify the way your terminal window appears on screen. Refer to Chapter 6, *Command Reference*, under the sections **Emulations** and **Desktop** for details.

Programmable Keys

There are two groups of programmable keys in *QmodemPro for Windows:* Remapped keyboard keys (KeyMaps) associated with terminal emulations, and Macro keys (at the bottom of the terminal window) associated with phonebook entries.

Emulation KeyMaps

Macros

Application examples

Menu Command Macros

Emulation KeyMaps

Up to 160 reassignments may be programmed for each terminal emulation, consisting of 12 function keys and 33 additional keys in normal, shift, alternate and control positions, along with keys on the numeric keypad in numlock and cursor key modes.

To create and edit KeyMaps, go to the **Options/Emulations** menu, and select the emulation you wish to edit. Then click the **Keyboard** button.

You will see a map of the keyboard. The programmable keys are highlighted. To edit a key, select the key with your mouse and click the left button.

A dialog box will pop up labeled with the name of the key you are editing. The text entry accepts character (text) mode as a default. To enter a string in decimal or hex, select the appropriate option, below the text entry window.

The **Options** button allows you to specify how several items should operate in both the remapping dialog as well as your terminal emulation when it is active. Each item is reviewed in the following discussion.

The **F10** key in Windows is used to activate the current Window menu bar. The first option allows you to retain this functionality or to remap **F10** to other emulation-specific functionality.

The **Numlock** key is used by the keyboard remapper to toggle the numpad of the keyboard on your screen (while you are remapping) between numbers and arrows. When this option is set to **Keypad Toggle** it will operate in this fashion, and will also pass this functionality to the terminal screen. If it is set to **Mapped**, it will not be available to you in the remapping window to toggle the numpad, but may be mapped to other functionality. If it is Mapped, the assigned function is active in the terminal screen, and the numpad cannot be toggled while in terminal mode.

The **Tab** key in Windows is used to move the current screen focus from one item to another. It is used in this manner in the keyboard remapper to move the highlight from key to key.

The **keyboard type** for the remapper can be changed from the 101 key style to the older 84 key style if needed.

To save your new key assignments, click the **Close** button.

For a complete list of Macro Commands, please see Macro System Functions.

Macros

Macros are assigned to buttons located at the bottom of the terminal window. Macros can also be activated using the numeric keys at the top of the keyboard along with the **[Alt]**, **[Alt][Shift]**, **[Alt][Ctrl]** and **[Alt][Shift][Ctrl]** keys. Up to 40 macros can thus be created.

To create and edit Macro keys, go to the **Options/Macro keys** menu.

To create a new macro file, click the **New** button. Then select the key you want to edit by clicking on the corresponding button in the dialog box, then fill in the contents of the macro in the appropriate box. Don't forget to **save** your changes.

The **Set Default** button assigns the current macro file as the default in the terminal screen, for new phonebook entries, and for phonebook entries that do not have their own macro assignments.

To edit a Macro key, load the macro file you want to edit by clicking the **Load** button and entering the file information in the common dialog box which pops up. As with the emulation KeyMaps dialog box, you can enter your data in character mode, hex or decimal.

For a complete list of Macro Commands, please see Macro System Functions.

Note that, unlike older versions of *Qmodem* and *QmodemPro* for DOS, the Macro System Functions cannot be used in the script language.

Macro System Functions

@ADDLFTOCR [on | off]

Same as menu command **Terminal/Add LF to CR**

@AUTOANSWER [on | off]

Same as menu command **Device/Auto Answer ON/OFF**

@BREAK

Same as menu command **Device/Break**

@CAPTURE [filename | off]

Same as menu command **File/Capture**

@CLS

Same as menu command **Terminal/Clear Screen**

@DEVICE [devicename]

Same as menu command **Device/Select**

@DEVICEFINETUNE [settings]

Same as menu command **Device/Fine Tune**

@DIAL [ENTRY | GROUP | SEARCH | MANUAL] string

Dials the entry specified on the command line, either explicitly by entry number, a group specified by name, by searching for text within phonebook entries, or by prompting for a phone number.

@DOORWAY [on | off]

Same as menu command **Emulation/Doorway Mode**

@DOWNLOAD [filename/dir [protocol]]

Downloads the specified file, using the specified protocol.

@DUPLEX [on | off]

Same as menu command **Device/Full Duplex**

@EDIT [filename]

Same as menu command **File/Edit a File**

@EMULATION [emulation]

Same as menu command **Terminal/Emulation**

@EXIT

Same as menu command **File/Exit**

@FAXAUTORECEIVE

Same as menu command **Fax/Auto Receive**

@FAXRECEIVE

Same as menu command **Fax/Receive**

@FAXRECEIVELOG

Same as menu command **Fax/Receive Log**

@FAXSEND

Same as menu command **Fax/Send**

@HANGUP

Same as menu command **Device/Hangup**

@HELP

Pops up the *QmodemPro for Windows* Help system in the Terminal Window. Same as menu command **Help/Contents**

@HOST

Same as menu command **Scripts/Host Mode Script**

@MANUALDIAL [number]

Same as menu command **Dial/Manual Dial**

@PASSWORD

Extracted from **Password** field in current phonebook entry. This only works for systems you dial from the phonebook.

@PHONEBOOK [phonebookname]

Same as Phonebook menu command **Phonebook/Open**

@PRINTER [on | off]

Same as menu command **File/Printer Toggle**

@QUICKLEARN [scriptname]

Same as menu command **QuickLearn.**

@RESETEMULATION

Same as menu command **Terminal/Reset emulation**

@RESETMODEM

Same as menu command **Device/Reset Modem**

@SCRIPT [scriptname]

Same as menu command **Scripts/Execute**

@SCRIPTCOMPILE [scriptname]

Same as menu command **Scripts/Compile**

@SCRIPTDEBUG [scriptname]

Same as menu command **Scripts/Debug**

@SCRIPTEDIT [filename]

Same as menu command **Scripts/Edit a Script**

@SCROLLBACK [on | off]

Same as menu command **Terminal/Scrollback**

@SCROLLBACKRECORD [on | off]

Same as menu command **Terminal/Scrollback Record**

@SHELL [command]

Same as menu command **File/Shell to DOS**

@SPLITSCREEN [on | off]

Same as menu command **Emulations/Split Screen**

@STAMP [string]

Records an entry to the **log file**.

@STRIPHIBIT [on | off]

Same as menu command **Emulations/8th Bit Strip**

@TRAPSCREEN [filename]

Same as menu command **Edit/Screen to File**

@UPLOAD [filenames [protocol]]

Uploads the file or files specified on the command line, using the specified non-batch or batch protocol.

@USERID

Extracted from **UserID** field in current phonebook entry. This only works for systems you dial from the phonebook.

@VIEW [filename]

Same as menu command **File/View a File**

@VIEWGIF [filenames]

Same as menu command **File/View a GIF File**

@XONXOFF [on | off]

Same as menu command **Device/Xon Xoff**

Application examples

Following are just a few examples of how Emulation KeyMaps and Macros can be used in *QmodemPro for Windows*.

Launching Off-Line Readers and other DOS applications

The macro @SHELL allows you to open a DOS window and run a DOS application such as an off-line mail reader with the touch of a key. An example of the @SHELL macro might look like

    @SHELL C:\OLX\OLX.EXE

This command runs Off-Line Xpress so you can read QWK mail packets. *QmodemPro for Windows* will change to the drive and subdirectory specified in the command line before executing the program. Substitute the path and command line for your own reader, if it is different from the example.

The next example starts Off-Line Xpress from a Windows .PIF file, create earlier in the .PIF editor. It uses whatever instructions the .PIF file contains:

    @SHELL OLX.PIF

The final example specifies an executable program name only, and is executed from the QMWIN subdirectory.

    @SHELL OLX.EXE

Type

    EXIT **[Enter]**

to close the DOS Shell window.

An Edit Macro

The @EDIT macro function allows you to call up the editor to edit or view a file. If a filename follows @EDIT, that file will be loaded into the editor immediately. If no filename has been specified, you will be prompted for one. For example:

    @EDIT

The Windows Common Dialog box will pop up to prompt for the filename to edit.

    @EDIT HOST.USR

This will edit the internal HOST user database file.

A View Macro

The @VIEW and @VIEWGIF macros allow you pop up the File Viewer and GIF Viewer with the file of your choice.

    @VIEW QMWIN.LOG

This loads the QMWIN.LOG file into the viewer, and is equivalent to the **File/Log View** menu command.

    @VIEWGIF MSI*.GIF

This loads all files matching the DOS Wildcard MSI*.GIF into the GIF viewer. Files are

displayed in succession, with a brief pause between each screen, like a slide show.

A Script Macro

The @SCRIPT macro function causes execution of the Script whose filename follows. The .SCR script file extension is not required. The @SCRIPT macro function and the script name must be separated by at least one space. For example:

```
@SCRIPT TEST1
```

This will call the script "TEST1.SCR". Because the extension ".SCR" was not included in the filename, it is added by default.

A LOG Macro

The @STAMP macro function causes the text following it to be recorded into the current LOG file. For example:

```
@STAMP This will be put in the LOG file.
```

The LOG file entry would look like this (except with the current time and date):

```
16:26:54 01-01-94 This will be put in the LOG file.
```

The time and date will only be included if it is enabled in **Options/Files/File Definitions**.

The @USERID and @PASSWORD Functions

Special system functions have been established for use in macro definition to pull information from the phonebook and other system areas. When used in a key macro definition, @USERID and @PASSWORD can be placed in the text to call functions that return the current phonebook entry's USERID and PASSWORD field. Whenever a connection is made, the data in these fields is available through the corresponding functions. Unlike all the other special macro commands that must appear first in the macro, the @PASSWORD and @USERID functions may be used anywhere within the macro text. For example, let's assume that JIM HARRER is my User ID, and the word "QMPRO!" is the password for the entry we have just connected to, and the following was coded for **[Alt][9]** in **Options/Macros**:

```
@USERID @PASSWORD
```

After the connection is made, pressing **[Alt][9]** would send the following:

```
JIM HARRER QMWIN!
```

Using these functions allows you to use the same macro key for each connection when your User ID and password are requested, as long as you have recorded this information in each dialing entry in the phonebook.

Note that although you can assign @USERID and @PASSWORD to an Emulation KeyMap or Macro, *QmodemPro for Windows* will only send the password during the actual connection. At other times no value is returned by these functions since they don't know which entry to use until a connection is made.

For a complete list of Macro Commands, please see Macro System Functions.

Menu Command Macros

Many of the more frequently used menu commands are already assigned to buttons on the Tool Bar. This provides a handy short-cut to the commands you use the most. You can also define macro buttons and function keys with menu commands you expect to use frequently.

These menu command macros have the additional advantage of being able to specify parameters. Here are some examples.

```
@EMULATION RIPscrip
```

This command changes the current terminal emulation to RIPscrip.

```
@DEVICEFINETUNE E 7 1
```

This command changes the current device settings on the fly to Even parity, 7 data bits and 1 stop bit.

```
@DEVICE USR HST DS
```

Meanwhile, this command loads a device called USR HST DS. This becomes the current device.

```
@DOORWAY
```

This command toggles **Doorway Mode**.

```
@RESETMODEM
```

This command sends the **modem initialization string** as defined in **Options/Device/Modem**, to the current device.

For a complete list of Macro Commands, please see Macro System Functions.

Special Control characters

QmodemPro for Windows has a special way of entering control characters in dialog boxes, macro commands, and scripts. Some control characters have no keyboard equivalents -- for instance the half second delay -- while others have keyboard equivalents which are reserved by Windows or *QmodemPro* for other uses. This includes the **[Enter]** and **[Esc]** keys, and the **[Ctrl]** key itself.

Control characters

The "^" caret or **[Shift][6]** character indicates that the character immediately following should be interpreted as a control character. To send a literal "^" character, follow it with a "|" vertical bar like this "^|".

The Carriage Return character

A carriage return is usually needed at the end of a modem dialing command. In *QmodemPro for Windows* we use the **[Ctrl][M]** to represent the **[Enter]** key. You would enter it like this:

```
^M
```

The Escape character

There may be times when you need to enter the **[Esc]** character in a script or macro, particularly when you call a Bulletin Board that uses a front-end mailer. The mailer usually prompts you to hit your **[Esc]** key twice to load the BBS. That's easy enough to do from the keyboard, but when you're trying to save that character in a macro, pressing your **[Esc]** key just closes the dialog box! The equivalent command code for the **[Esc]** is "^[".

Adding a half second pause

A special control key combination is used to pause for half a second before executing the next part of your command. Use the sequence "^~" as many times as necessary to build a delay into your macro.

Adding a two second pause

While the previous example works for sending modem commands in a macro, there is another way of sending a delay directly to the modem during a dialing command. Most modems interpret the comma as "wait two seconds before proceeding with the rest of the command".

Character translation

Translation may be necessary when communicating with another computers whose ASCII codes (the computer's alphabet) do not match those used by your PC. All 256 characters of the IBM ASCII character set are available for translation.

Translation tables are associated with Terminal Emulations, and can be edited from the **Options/Emulations/Translate** menu. There are separate translations for incoming and outgoing data. The Input table controls which incoming characters are passed unchanged, and which are translated to other characters. The Output table controls which characters are sent out your modem to the other system.

You can strip an incoming character completely by translating the character to 0 in the outgoing translate table.

The original character is shown to the left of each list, while the result of the translation is shown on the right.

The **Edit Format** options near the bottom of the **Translations** dialog box let you view and edit your translation tables in ASCII (character mode), decimal (base 10 numeric representation of characters) or hexadecimal (base 16 alphanumeric representation of characters).

The procedure for editing a character translation is the same for incoming and outgoing translations. First, double-click on the character you want to translate. A dialog box will pop up, allowing you to enter the new value.

Text entry

Most characters can be entered in text mode by typing the corresponding letter on the keyboard. You may prefer to change edit modes and enter control characters and high ASCII characters as their decimal or hex equivalents.

Normal characters can be entered into this field using the keyboard. For control characters, use the **[Shift][6]** key to produce the "^" control symbol, then the character. For instance, to enter **[Ctrl][B]**, type "^B". Holding down the **[Ctrl]** and typing a **[B]** will not work.

Decimal and Hex entry

High-ASCII characters that have no keyboard equivalent, for instance foreign accents and line drawing characters, must be entered in decimal or hex mode. You can find the decimal and hex values of the characters you want to translate by highlighting the desired character with one mouse click, then switching Edit Format from ASCII to Decimal or Hex. Type in the appropriate three digit number (for Decimal) or two digit alphanumeric (for Hex).

Use the **OK** button to save your changes and return to the **Options/Emulations** menu, or press the **[Esc]** key to abandon your changes.

Other communication devices

QmodemPro for Windows is able to support other kinds of connections besides an ordinary modem connected to a serial port on your PC. In most cases, these alternative devices are installed as part of your Windows configuration.

Since the installation and usage of these alternate devices varies from one product to another, we will discuss them only briefly here. For more details on how to install and use other communication devices, refer to the documentation supplied with the communication hardware, local area network, or other device you would like to use with *QmodemPro for Windows.*

Below are a number of possible configurations.

Network modems

One of the most common ways to make use of a modem on a network is through use of Interrupt 14 to redirect output to a network device. INT 14 is internally supported in *QmodemPro for Windows* without the need for additional software. If your network modem or communications server supports INT 14 you can select any of 4 channels, designated as Com1: Int 14 through Com4: Int 14. These are selectable in the same manner as standard serial ports using **Options/Device/Hardware** and choosing the appropriate port.

In situations using protocols other than INT 14 you may need to install network device drivers in your workstation running Windows. If the communications driver actually replaces a standard serial driver, *QmodemPro for Windows* will be able to recognize and use modems referenced by these device drivers. In many cases these devices, once they are set up and configured, are addressed by Windows applications as regular COM ports. The physical COM port in question need not necessarily be present in your machine -- Windows looks after redirecting serial input and output to the proper device, whether it is connected to your PC or one located elsewhere on the LAN.

Your network administrator is your best source of information on the correct configuration for modems attached in this manner.

DigiBoard and other intelligent multi-port serial cards

Intelligent multi-port serial cards work well with *QmodemPro for Windows*, providing the Windows drivers for such cards are properly installed. An example is the intelligent DigiBoard card, from DIGI International. The card is supplied with a Windows driver which is installed in the same way as other Windows applications.

When the driver is installed, the configuration menu allows you to map ports on the card to Windows COM ports. As with LAN modems, the physical COM port in question need not necessarily be present in your machine -- Windows looks after redirecting serial input and output to the proper port on your multiport card.

Direct Connection via Serial Cable

At times, you may want to connect two computers through their serial ports using a serial cable and a null modem. This is a useful way to copy files from a laptop PC to a desktop machine, or from a standalone word processing machine to your PC, or log into a mini or mainframe computer through a serial connection.

To set up a machine for direct connection, go to the **Options/Device** menu and create a new device called "Direct Connection". Select this new device, and click the **<u>Modem</u>** button. In the field for **Connect Message**, enter the word DIRECT. Leave all the other fields blank: Modem init, Dial Prefix, Dial Suffix, the various connect messages, and so forth.

The next step is to link this device to a Phonebook entry. Go to the Phonebook, select **Add**, and select "Direct Connection" as the device to use. Leave the phone number fields blank.

When you dial this entry, it will report an immediate "DIRECT connection established" message, and exit back to the terminal window. It will also run any script attached to this entry to automate logons or any other operation you need to perform.

ADDS VP60

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^B1^M
F2	Function 2 (F2)	^B2^M
F3	Function 3 (F3)	^B3^M
F4	Function 4 (F4)	^B4^M
F5	Function 5 (F5)	^B5^M
F6	Function 6 (F6)	^B6^M
F7	Function 7 (F7)	^B7^M
F8	Function 8 (F8)	^B8^M
Backspace	Backspace	^H
Enter	Return	^M
Gray Insert	Insert Character	^[F
Gray Delete	Delete Character	^[E
Gray Home	Home Cursor	^A
Gray End	Erase End of Line	^[K
Gray Up	Cursor Up	^Z
Gray Down	Cursor Down	^J
Gray Left	Cursor Back	^U
Gray Right	Cursor Forward	^F
Numlock	Cursor Down (Line Feed)	^J
Pad /	Cursor Back	^U
Pad *	Cursor Forward	^F
Pad -	Cursor Up	^Z
Pad Enter	Enter	^M
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Delete Character	^[E
Pad Insert	Insert Character	^[F
Pad End	Erase End of Line	^[K
Pad Down	Cursor Down	^J
Pad Pagedown	3	3
Pad Left	Cursor Back	^U
Pad Clear	5	5
Pad Right	Cursor Forward	^F
Pad Home	Cursor Home	^A
Pad Up	Cursor Up	^Z
Pad Pageup	9	9
Shift+Tab	Back Tab	^[O
Shift+F1	Shift-Function 1 (Shift F1)	^B!^M
Shift+F2	Shift-Function 2 (Shift F2)	^B"^M
Shift+F3	Shift-Function 3(Shift F3)	^B#^M
Shift+F4	Shift-Function 4 (Shift F4)	^B$^M
Shift+F5	Shift-Function 5 (Shift F5)	^B%^M

Shift+F6	Shift-Function 6 (Shift F6)	^B&^M
Shift+F7	Shift-Function 7 (Shift F7)	^B'^M
Shift+F8	Shift-Function 8 (Shift F8)	^B(^M
Shift+Enter	Enter	^M
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End	1	1
Shift+Pad Down	2	3
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	7	7
Shift+Pad Up	8	8
Shift+Pad Pageup	9	9
Control+F5	Insert Line	^[M
Control+F6	Delete Line	^[I
Control+F7	Insert Character	^[F
Control+F8	Delete Character	^[E
Control+F9	Erase All (E All)	^L
Control+F10	Erase End of Field/Line	^[K
Control+Backspace	Ctrl-Backspace	Δ

ADM 3A

QmodemPro Key	Terminal Key	Characters Sent
Tab	Horizontal Tab	^I
Backspace	Backspace	^H
Enter	Enter	^M
Gray Delete	Character Delete	Δ
Gray Home	Cursor Home	^^
Gray End	Erase End of Line	^[K
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^J
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7

Pad 8	8	8
Pad 9	9	9
Pad Delete	Character Delete	Δ
Pad End	Erase End of Line	^[K
Pad Down	Cursor Down	^J
Pad Left	Cursor Left	^H
Pad Right	Cursor Right	^L
Pad Home	Cursor Home	^^
Pad Up	Cursor Up	^K
Shift+Enter	Enter	^M
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End	1	1
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	7	7
Shift+Pad Up	8	8
Shift+Pad Pageup	9	9
Control+Backspace	Character Delete	Δ

ANSI

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Program Function 1 (PF1)	^[OP
F2	Program Function 2 (PF2)	^[OQ
F3	Program Function 3 (PF3)	^[OR
F4	Program Function 4 (PF4)	^[OS
Backspace	Backspace	^H
Enter	Return	^M
Gray Delete	Delete Character	Δ
Gray Home	Home Cursor	^[[H
Gray End	Erase End of Line	^[[K
Gray Pageup	Mapped by QmodemPro	@UPLOAD
Gray Pagedown	Mapped by QmodemPro	@DOWNLOAD
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5

Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Delete Character	Δ
Pad End	Erase End of Line	^[[K
Pad Down	Cursor Down	^[[B
Pad Pagedown	Mapped by QmodemPro	@DOWNLOAD
Pad Left	Cursor Left	^[[D
Pad Right	Cursor Right	^[[C
Pad Home	Home Cursor	^[[H
Pad Up	Cursor Up	^[[A
Pad Pageup	Mapped by QmodemPro	@UPLOAD
Shift+Backspace	Delete Character	Δ
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete Character	Δ
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	-
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End	1	1
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	7	7
Shift+Pad Up	8	9
Shift+Pad Pageup	9	9
Control+Backspace	0	^X
Control+Enter	Enter	^M
Control+Gray Home	Insert Line	^[[L
Control+Gray Pageup	Delete Line	^[[M
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad .	Keypad Application PERIOD	^[On
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application PERIOD	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq

QmodemPro Key	Terminal Key	Characters Sent
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot
Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 9	^[Oy
Shift+Control+Enter	Enter	^M
Shift+Control+Pad Enter	Enter	^M
Shift+Control+Pad Home	Insert LIne	^[[L

Avatar

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Program Function 1 (PF1)	^[OP
F2	Program Function 2 (PF2)	^[OQ
F3	Program Function 3 (PF3)	^[OR
F4	Program Function 4 (PF4)	^[OS
Backspace	Backspace	^H
Enter	Enter	^M
Gray Delete	Delete Character	Δ
Gray Home	Home Cursor	^[[H
Gray End	Erase End of Line	^[[K
Gray Pageup	Mapped by QmodemPro	@UPLOAD
Gray Pagedown	Mapped by QmodemPro	@DOWNLOAD
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Delete Character	Δ
Pad End	Erase End of Line	^[[K
Pad Down	Cursor Down	^[[B
Pad Pagedown	Mapped by QmodemPro	@DOWNLOAD
Pad Left	Cursor Left	^[[D
Pad Right	Cursor Right	^[[C
Pad Home	Home Cursor	^[[H
Pad Up	Cursor Up	^[[A

Pad Pageup	Mapped by QmodemPro	@UPLOAD
Shift+Backspace	Delete Character	Δ
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete Character	Δ
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End	1	1
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	7	7
Shift+Pad Up	8	9
Shift+Pad Pageup	9	9
Control+Backspace	Ctrl-Backspace	^X
Control+Enter	Enter	^M
Control+Gray Home	Insert Line	^[[L
Control+Gray Pageup	Delete Line	^[[M
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad .	Keypad Application PERIOD	^[On
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application PERIOD	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot
Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 9	^[Oy
Shift+Control+Enter	Enter	^M
Shift+Control+Pad Enter	Enter	^M
Shift+Control+Pad Home	Insert Line	^[[L

Debug Ascii

This is a custom emulation for debugging connections. It resembles the TTY emulation with the exception that ASCII codes 0-31 and 255 are displayed in mnemonics surrounded by square brackerts. The list is as follows:

CODE	MNEMONIC	CODE	MNEMONIC	CODE	MNEMONIC
0	[NUL]	11	[VT]	22	[SYN]
1	[SOH]	12	[FF]	23	[ETB]
2	[STX]	13	[CR]	24	[CAN]
3	[ETX]	14	[SO]	25	[EN]
4	[EOT]	15	[SI]	26	[SUB]
5	[ENQ]	16	[DLE]	27	[ESC]
6	[ACK]	17	[DC1]	28	[FS]
7	[BEL]	18	[DC2]	29	[GS]
8	[BS]	19	[DC3]	30	[RS]
9	[HT]	20	[DC4]	31	[US]
10	[LF]	21	[NAK]	255	[0xFF]

Debug Hex

This is also a custom Emulation for debugging connections. All data received is displayed as hexadecimal numbers between square brackets.

A sample looks like this:

```
[61][74][0D][0A][4F][4B][0D][0A]
```

The above is the result of typing AT **[Enter]** and the modem responding with OK.

DG 100

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (Cmd-1)	^^q
F2	Function 2 (Cmd-2)	^^r
F3	Function 3 (Cmd-3)	^^s
F4	Function 4 (Cmd-4)	^^t
F5	Function 5 (Cmd-5)	^^u
F6	Function 6 (Cmd-6)	^^v
F7	Function 7 (Cmd-7)	^^w
F8	Function 8 (Cmd-8)	^^x
F9	Function 9 (Cmd-9)	^^y
F10	Function 10 (Cmd-10)	^^z
Backspace	Carriage Return	^M
Enter	New Line	^J
Gray Delete	Delete	Δ
Gray Home	Home Cursor	^H
Gray Up	Cursor Up	^W
Gray Down	Cursor Down	^Z
Gray Left	Cursor Left	^Y
Gray Right	Cursor Right	^X
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	New Line	^J
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Delete	Δ
Pad Down	Cursor Down	^Z
Pad Left	Cursor Left	^Y
Pad Right	Cursor Right	^X
Pad Home	Home Cursor	^H
Pad Up	Cursor Up	^W
Shift+F1	Shift-Function 1 (Cmd-Shift-1)	^^a
Shift+F2	Shift-Function 2 (Cmd-Shift-2)	^^b
Shift+F3	Shift-Function 3 (Cmd-Shift-3)	^^c
Shift+F4	Shift-Function 4 (Cmd-Shift-4)	^^d
Shift+F5	Shift-Function 5 (Cmd-Shift-5)	^^e
Shift+F6	Shift-Function 6 (Cmd-Shift-6)	^^f
Shift+F7	Shift-Function 7 (Cmd-Shift-7)	^^g
Shift+F8	Shift-Function 8 (Cmd-Shift-8)	^^h
Shift+F9	Shift-Function 9 (Cmd-Shift-9)	^^i
Shift+F10	Shift-Function 10 (Cmd-Shift-10)	^^j

QmodemPro Key	Terminal Key	Characters Sent
Control+Enter	Carriage Return	^M
Control+Pad Enter	Carriage Return	^M
Control+Pad .	Delete	Δ
Control+Pad 2	Cursor Down	^Z
Control+Pad 4	Cursor Left	^Y
Control+Pad 6	Cursor Right	^X
Control+Pad 7	Home Cursor	^H
Control+Pad 8	Cursor Up	^W
Control+Pad Delete	.	.
Control+Pad Insert	0	0
Control+Pad End	1	1
Control+Pad Down	2	2
Control+Pad Pagedown	3	3
Control+Pad Left	4	4
Control+Pad Clear	5	5
Control+Pad Right	6	6
Control+Pad Home	7	7
Control+Pad Up	8	8
Control+Pad Pageup	9	9

DG 200

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1	^^q
F2	Function 2	^^r
F3	Function 3	^^s
F4	Function 4	^^t
F5	Function 5	^^u
F6	Function 6	^^v
F7	Function 7	^^w
F8	Function 8	^^x
F9	Function 9	^^y
F10	Function 10	^^z
Backspace	Carriage Return	^M
Enter	New Line	^J
Gray Delete	Delete	Δ
Gray Home	Home	^H
Gray Up	Cursor Up	^W
Gray Down	Cursor Down	^Z
Gray Left	Cursor Left	^Y
Gray Right	Cursor Right	^X
Pad /	Erase Page	^L
Pad *	*	*
Pad -	Erase End of Line	^K
Pad +	+	+
Pad Enter	New Line	^J
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6		6

Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Delete	Δ
Pad End	C3	^^^|
Pad Down	Cursor Down	^Z
Pad Pagedown	C4	^^_
Pad Left	Cursor Left	^Y
Pad Clear	Home	^H
Pad Right	Cursor Right	^X
Pad Home	C1	^^\
Pad Up	Cursor Up	^W
Pad Pageup	C2	^^]
Shift+F1	Shift-Function 1	^^a
Shift+F2	Shift-Function 2	^^b
Shift+F3	Shift-Function 3	^^c
Shift+F4	Shift-Function 4	^^d
Shift+F5	Shift-Function 5	^^e
Shift+F6	Shift-Function 6	^^f
Shift+F7	Shift-Function 7	^^g
Shift+F8	Shift-Function 8	^^h
Shift+F9	Shift-Function 9	^^i
Shift+F10	Shift-Function 10	^^j
Shift+Gray Up	Shift-Cursor Up	^^^W
Shift+Gray Down	Shift-Cursor Down	^^^Z
Shift+Gray Left	Shift-Cursor Left	^^^Y
Shift+Gray Right	Shift-Cursor Right	^^^X
Shift+Pad End	Shift-C3	^^Z
Shift+Pad Down	Shift-Cursor Down	^^^Z
Shift+Pad Pagedown	Shift-C4	^^[
Shift+Pad Left	Shift-Cursor Left	^^^Y
Shift+Pad Clear	Shift-Home	^^^H
Shift+Pad Right	Shift-Cursor Right	^^^X
Shift+Pad Home	Shift-C1	^^X
Shift+Pad Up	Shift-Cursor Up	^^^W
Shift+Pad Pageup	Shift-C2	^^Y
Control+F1	Ctrl-Function 1	^^1
Control+F2	Ctrl-Function 2	^^2
Control+F3	Ctrl-Function 3	^^3
Control+F4	Ctrl-Function 4	^^4
Control+F5	Ctrl-Function 5	^^5
Control+F6	Ctrl-Function 6	^^6
Control+F7	Ctrl-Function 7	^^7
Control+F8	Ctrl-Function 8	^^8
Control+F9	Ctrl-Function 9	^^9
Control+F10	Ctrl-Function 10	^^:
Control+Enter	Carriage Return	^M
Control+Pad Enter	Carriage Return	^M
Control+Pad .	Delete	Δ
Control+Pad 2	Cursor Down	^Z
Control+Pad 4	Cursor Left	^Y
Control+Pad 6	Cursor Right	^X
Control+Pad 7	Home	^H
Control+Pad 8	Cursor Up	^W
Control+Pad Delete	.	.
Control+Pad Insert	0	0

Control+Pad End	1	1
Control+Pad Down	2	2
Control+Pad Pagedown	3	3
Control+Pad Left	4	4
Control+Pad Clear	5	5
Control+Pad Right	6	6
Control+Pad Home	7	7
Control+Pad Up	8	8
Control+Pad Pageup	9	9
Shift+Control+F1	Shift Ctrl-Function 1	^^!
Shift+Control+F2	Shift Ctrl-Function 2	^^"
Shift+Control+F3	Shift Ctrl-Function 3	^^#
Shift+Control+F4	Shift Ctrl-Function 4	^^$
Shift+Control+F5	Shift Ctrl-Function 5	^^%
Shift+Control+F6	Shift Ctrl-Function 6	^^&
Shift+Control+F7	Shift Ctrl-Function 7	^^'
Shift+Control+F8	Shift Ctrl-Function 8	^^(
Shift+Control+F9	Shift Ctrl-Function 9	^^)
Shift+Control+F10	Shift Ctrl-Function 10	^^*

DG 210

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1	^^q
F2	Function 2	^^r
F3	Function 3	^^s
F4	Function 4	^^t
F5	Function 5	^^u
F6	Function 6	^^v
F7	Function 7	^^w
F8	Function 8	^^x
F9	Function 9	^^y
F10	Function 10	^^z
Backspace	Carriage Return	^M
Enter	New Line	^J
Gray Delete	Delete	Δ
Gray Home	Home	^H
Gray Up	Cursor Up	^W
Gray Down	Cursor Down	^Z
Gray Left	Cursor Left	^Y
Gray Right	Cursor Right	^X
Pad /	Erase Page	^L
Pad *	*	*
Pad -	Erase End of Line	^K
Pad +	+	+
Pad Enter	New Line	^J
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6		6

Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Delete	Δ
Pad End	C3	^^^|
Pad Down	Cursor Down	^Z
Pad Pagedown	C4	^^_
Pad Left	Cursor Left	^Y
Pad Clear	Home	^H
Pad Right	Cursor Right	^X
Pad Home	C1	^^\
Pad Up	Cursor Up	^W
Pad Pageup	C2	^^]
Shift+F1	Shift-Function 1	^^a
Shift+F2	Shift-Function 2	^^b
Shift+F3	Shift-Function 3	^^c
Shift+F4	Shift-Function 4	^^d
Shift+F5	Shift-Function 5	^^e
Shift+F6	Shift-Function 6	^^f
Shift+F7	Shift-Function 7	^^g
Shift+F8	Shift-Function 8	^^h
Shift+F9	Shift-Function 9	^^i
Shift+F10	Shift-Function 10	^^j
Shift+Gray Up	Shift-Cursor Up	^^^W
Shift+Gray Down	Shift-Cursor Down	^^^Z
Shift+Gray Left	Shift-Cursor Left	^^^Y
Shift+Gray Right	Shift-Cursor Right	^^^X
Shift+Pad End	Shift-C3	^^Z
Shift+Pad Down	Shift-Cursor Down	^^^Z
Shift+Pad Pagedown	Shift-C4	^^[
Shift+Pad Left	Shift-Cursor Left	^^^Y
Shift+Pad Clear	Shift-Home	^^^H
Shift+Pad Right	Shift-Cursor Right	^^^X
Shift+Pad Home	Shift-C1	^^X
Shift+Pad Up	Shift-Cursor Up	^^^W
Shift+Pad Pageup	Shift-C2	^^Y
Control+F1	Ctrl-Function 1	^^1
Control+F2	Ctrl-Function 2	^^2
Control+F3	Ctrl-Function 3	^^3
Control+F4	Ctrl-Function 4	^^4
Control+F5	Ctrl-Function 5	^^5
Control+F6	Ctrl-Function 6	^^6
Control+F7	Ctrl-Function 7	^^7
Control+F8	Ctrl-Function 8	^^8
Control+F9	Ctrl-Function 9	^^9
Control+F10	Ctrl-Function 10	^^:
Control+Enter	Carriage Return	^M
Control+Pad Enter	Carriage Return	^M
Control+Pad .	Delete	Δ
Control+Pad 2	Cursor Down	^Z
Control+Pad 4	Cursor Left	^Y
Control+Pad 6	Cursor Right	^X
Control+Pad 7	Home	^H
Control+Pad 8	Cursor Up	^W
Control+Pad Delete	.	.
Control+Pad Insert	0	0

Control+Pad End	1	1
Control+Pad Down	2	2
Control+Pad Pagedown	3	3
Control+Pad Left	4	4
Control+Pad Clear	5	5
Control+Pad Right	6	6
Control+Pad Home	7	7
Control+Pad Up	8	8
Control+Pad Pageup	9	9
Shift+Control+F1	Shift Ctrl-Function 1	^^!
Shift+Control+F2	Shift Ctrl-Function 2	^^"
Shift+Control+F3	Shift Ctrl-Function 3	^^#
Shift+Control+F4	Shift Ctrl-Function 4	^^$
Shift+Control+F5	Shift Ctrl-Function 5	^^%
Shift+Control+F6	Shift Ctrl-Function 6	^^&
Shift+Control+F7	Shift Ctrl-Function 7	^^'
Shift+Control+F8	Shift Ctrl-Function 8	^^(
Shift+Control+F9	Shift Ctrl-Function 9	^^)
Shift+Control+F10	Shift Ctrl-Function 10	^^*

Hazeltine 1500

No key substitution performed.

Heath 19

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Program Function 1 (PF1)	^[s
F2	Program Function 2 (PF2)	^[t
F3	Program Function 3 (PF3)	^[u
F4	Program Function 4 (PF4)	^[v
F5	Program Function 5 (PF5)	^[w
F6	Program Function 6 (PF6)	^[P
F7	Program Function 7 (PF7)	^[Q
F8	Program Function 8 (PF8)	^[R
F9	Program Function 9 (PF9)	^[0
F10	Program Function 10 (PF10)	^[1
Backspace	Backspace	^H
Enter	Enter	^M
Gray Delete	Delete Character	Δ
Gray Home	Home Cursor	^[Y_
Gray End	Erase ENd of Line	^[T
Gray Up	Cursor Up	^[A
Gray Down	Cursor Down	^[B
Gray Left	Cursor Left	^[D
Gray Right	Cursor Right	^[C
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Del	Δ
Pad End	Erase End of Line	^[T
Pad Down	Cursor Down	^[B
Pad Left	Cursor Left	^[D
Pad Right	Cursor Right	^[C
Pad Home	Home Cursor	^[Y_
Pad Up	Cursor Up	^[A
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End	1	1

Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	7	7
Shift+Pad Up	8	8
Shift+Pad Pageup	9	9
Control+Backspace	Del	Δ
Control+Gray Home	Insert Line	^[L
Control+Gray Pageup	Delete Line	^[M
Control+Gray Pagedown	Clear Screen	^[E
Control+Pad Pagedown	Clear Screen	^[E
Control+Pad Home	Insert Line	^[L
Control+Pad Pageup	Delete Line	^[M

IBM 3101

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	PF1	^[a^M
F2	PF2	^[b^M
F3	PF3	^[c^M
F4	PF4	^[d^M
F5	PF5	^[e^M
F6	PF6	^[f^M
F7	PF7	^[g^M
F8	PF8	^[h^M
Backspace	Backspace	^H
Enter	Enter	^M
Gray Home	Cursor Home	^[H
Gray End	Erase Input	^[K
Gray Up	Cursor Up	^[A
Gray Down	Cursor Down	^[B
Gray Left	Cursor Left	^[D
Gray Right	Cursor Right	^[C
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad End	Erase Input	^[K
Pad Down	Cursor Down	^[B
Pad Left	Cursor Left	^[D

Pad Right	Cursor Right	^[C
Pad Home	Home Cursor	^[H
Pad Up	Cursor Up	^[A
Shift+F1	Erase EOL/EOF	^[I
Shift+F2	Erase Input	^[K
Shift+F3	Erase EOS	^[J
Shift+F4	Clear	^[L
Shift+Enter	Enter	^M
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End	1	1
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	7	7
Shift+Pad Up	8	8
Shift+Pad Pageup	9	9
Control+Backspace	Delete Character	Δ
Control+Gray Home	Clear	^[L
Control+Gray End	Erase EOL/EOF	^[I
Control+Gray Pagedown	Erase EOS	^[J
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	-	-
Control+Pad +	+	+
Control+Pad .	.	.
Control+Pad 1	PF1	^[a^M
Control+Pad 2	PF2	^[b^M
Control+Pad 3	PF3	^[c^M
Control+Pad 4	PF4	^[d^M
Control+Pad 5	PF5	^[e^M
Control+Pad 6	PF6	^[f^M
Control+Pad 7	PF7	^[g^M
Control+Pad 8	PF8	^[h^M
Control+Pad End	Erase EOL/EOF	^[I
Control+Pad Pagedown	Erase EOS	^[J
Control+Pad Home	Clear	^[L

RIPscrip

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Program Function 1 (PF1)	^[OP
F2	Program Function 2 (PF2)	^[OQ
F3	Program Function 3 (PF3)	^[OR
F4	Program Function 4 (PF4)	^[OS
Backspace	Backspace	^H
Enter	Return	^M

Gray Delete	Delete Character	Δ
Gray Home	Home Cursor	^[[H
Gray End	Erase End of Line	^[[K
Gray Pageup	Mapped by QmodemPro	@UPLOAD
Gray Pagedown	Mapped by QmodemPro	@DOWNLOAD
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Delete Character	Δ
Pad End	Erase End of Line	^[[K
Pad Down	Cursor Down	^[[B
Pad Pagedown	Mapped by QmodemPro	@DOWNLOAD
Pad Left	Cursor Left	^[[D
Pad Right	Cursor Right	^[[C
Pad Home	Home Cursor	^[[H
Pad Up	Cursor Up	^[[A
Pad Pageup	Mapped by QmodemPro	@UPLOAD
Shift+Backspace	Delete Character	Δ
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete Character	Δ
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	-
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End	1	1
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	7	7
Shift+Pad Up	8	9
Shift+Pad Pageup	9	9
Control+Backspace	0	^X
Control+Enter	Enter	^M
Control+Gray Home	Insert Line	^[[L

Control+Gray Pageup	Delete Line	^[[M
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad .	Keypad Application PERIOD	^[On
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application PERIOD	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot
Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 9	^[Oy
Shift+Control+Enter	Enter	^M
Shift+Control+Pad Enter	Enter	^M
Shift+Control+Pad Home	Insert LIne	^[[L

TTY

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
Backspace	Backspace	^H
Enter	Enter	^M
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9

TVI 910

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M
F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Function 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M
F8	Function 8 (F8)	^AG^M
F9	Function 9 (F9)	^AH^M
F10	Function 10 (F10)	^AI^M
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Character Insert	^[Q
Gray Delete	Character Delete	^[W
Gray Home	Home Cursor	^^
Gray End	Erase End of Line	^[T
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^V
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Character Delete	^[W
Pad Insert	Character Insert	^[Q
Pad End	Erase End of Line	^[T
Pad Down	Cursor Down	^V
Pad Left	Cursor Left	^H
Pad Right	Cursor Right	^L
Pad Home	Home Cursor	^^
Pad Up	Cursor Up	^K
Shift+Tab	Back Tab	^[I
Shift+F1	Function 12 (F12)	^A`^M
Shift+F2	Function 13 (F13)	^Aa^M
Shift+F3	Function 14 (F14)	^Ab^M
Shift+F4	Function 15 (F15)	^Ac^M
Shift+F5	Function 16 (F16)	^Ad^M

QmodemPro Key	Terminal Key	Characters Sent
Shift+F6	Function 17 (F17)	^Ae^M
Shift+F7	Function 18 (F18)	^Af^M
Shift+F8	Function 19 (F19)	^Ag^M
Shift+F9	Function 20(F20)	^Ah^M
Shift+F10	Function 21 (F21)	^Ai^M
Shift+Gray Delete	Line Delete	^[R
Shift+Gray Home	Clear Page	^[{
Shift+Gray End	Erase End of Page	^[Y
Shift+Gray Pageup	Line Insert	^[E
Shift+Gray Up	Shift Cursor Up	^j
Shift+Gray Down	Shift Cursor Down	^J
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End		^[Y
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	Clear Page	^[{
Shift+Pad Up	8	8
Shift+Pad Pageup	9	9
Control+F7	Character Insert	^[Q
Control+F8	Character Delete	^[W
Control+F9	Line Insert	^[E
Control+F10	Line Delete	^[R
Control+Backspace	Delete Character	Δ
Control+Gray Delete	Shift Character Delete	^[r
Control+Gray Home	Clear All	^Z
Control+Pad Delete	Line Delete	^[R
Control+Pad Insert	Line Insert	^[E
Control+Pad Home	Clear All	^Z
Shift+Control+F7	Shift Character Insert	^[q
Shift+Control+F8	Shift Character Delete	^[r
Shift+Control+F9	Shift Line Insert	^[N
Shift+Control+F10	Shift Line Delete	^[O
Shift+Control+Pad End	1	1
Shift+Control+Pad Home	7	7

TVI 912

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M
F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Function 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M

F8	Function 8 (F8)	^AG^M
F9	Function 9 (F9)	^AH^M
F10	Function 10 (F10)	^AI^M
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Character Insert	^[Q
Gray Delete	Character Delete	^[W
Gray Home	Home Cursor	^^
Gray End	Erase End of Line	^[T
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^V
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Character Delete	^[W
Pad Insert	Character Insert	^[Q
Pad End	Erase End of Line	^[T
Pad Down	Cursor Down	^V
Pad Left	Cursor Left	^H
Pad Right	Cursor Right	^L
Pad Home	Home Cursor	^^
Pad Up	Cursor Up	^K
Shift+Tab	Back Tab	^[I
Shift+F1	Function 12 (F12)	^A`^M
Shift+F2	Function 13 (F13)	^Aa^M
Shift+F3	Function 14 (F14)	^Ab^M
Shift+F4	Function 15 (F15)	^Ac^M
Shift+F5	Function 16 (F16)	^Ad^M
Shift+F6	Function 17 (F17)	^Ae^M
Shift+F7	Function 18 (F18)	^Af^M
Shift+F8	Function 19 (F19)	^Ag^M
Shift+F9	Function 20(F20)	^Ah^M
Shift+F10	Function 21 (F21)	^Ai^M
Shift+Gray Delete	Line Delete	^[R
Shift+Gray Home	Clear Page	^[{
Shift+Gray End	Erase End of Page	^[Y
Shift+Gray Pageup	Line Insert	^[E
Shift+Gray Up	Shift Cursor Up	^j
Shift+Gray Down	Shift Cursor Down	^J
Shift+Pad /	/	/
Shift+Pad *	*	*

Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End		^[Y
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	Clear Page	^[{
Shift+Pad Up	8	8
Shift+Pad Pageup	9	9
Control+F7	Character Insert	^[Q
Control+F8	Character Delete	^[W
Control+F9	Line Insert	^[E
Control+F10	Line Delete	^[R
Control+Backspace	Delete Character	Δ
Control+Gray Delete	Shift Character Delete	^[r
Control+Gray Home	Clear All	^Z
Control+Pad Delete	Line Delete	^[R
Control+Pad Insert	Line Insert	^[E
Control+Pad Home	Clear All	^Z
Shift+Control+F7	Shift Character Insert	^[q
Shift+Control+F8	Shift Character Delete	^[r
Shift+Control+F9	Shift Line Insert	^[N
Shift+Control+F10	Shift Line Delete	^[O
Shift+Control+Pad End	1	1
Shift+Control+Pad Home	7	7

TVI 920

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M
F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Function 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M
F8	Function 8 (F8)	^AG^M
F9	Function 9 (F9)	^AH^M
F10	Function 10 (F10)	^AI^M
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Character Insert	^[Q
Gray Delete	Character Delete	^[W
Gray Home	Home Cursor	^^
Gray End	Erase End of Line	^[T
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^V
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L

Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Character Delete	^[W
Pad Insert	Character Insert	^[Q
Pad End	Erase End of Line	^[T
Pad Down	Cursor Down	^V
Pad Left	Cursor Left	^H
Pad Right	Cursor Right	^L
Pad Home	Home Cursor	^^
Pad Up	Cursor Up	^K
Shift+Tab	Back Tab	^[I
Shift+F1	Function 12 (F12)	^A`^M
Shift+F2	Function 13 (F13)	^Aa^M
Shift+F3	Function 14 (F14)	^Ab^M
Shift+F4	Function 15 (F15)	^Ac^M
Shift+F5	Function 16 (F16)	^Ad^M
Shift+F6	Function 17 (F17)	^Ae^M
Shift+F7	Function 18 (F18)	^Af^M
Shift+F8	Function 19 (F19)	^Ag^M
Shift+F9	Function 20(F20)	^Ah^M
Shift+F10	Function 21 (F21)	^Ai^M
Shift+Gray Delete	Line Delete	^[R
Shift+Gray Home	Clear Page	^[{
Shift+Gray End	Erase End of Page	^[Y
Shift+Gray Pageup	Line Insert	^[E
Shift+Gray Up	Shift Cursor Up	^[j
Shift+Gray Down	Shift Cursor Down	^J
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End		^[Y
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	Clear Page	^[{
Shift+Pad Up	8	8

Shift+Pad Pageup	9	9
Control+F7	Character Insert	^[Q
Control+F8	Character Delete	^[W
Control+F9	Line Insert	^[E
Control+F10	Line Delete	^[R
Control+Backspace	Delete Character	Δ
Control+Gray Delete	Shift Character Delete	^[r
Control+Gray Home	Clear All	^Z
Control+Pad Delete	Line Delete	^[R
Control+Pad Insert	Line Insert	^[E
Control+Pad Home	Clear All	^Z
Shift+Control+F7	Shift Character Insert	^[q
Shift+Control+F8	Shift Character Delete	^[r
Shift+Control+F9	Shift Line Insert	^[N
Shift+Control+F10	Shift Line Delete	^[O
Shift+Control+Pad End	1	1
Shift+Control+Pad Home	7	7

TVI 922

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	PF1	^[OP
F2	PF2	^[OQ
F3	PF3	^[OR
F4	PF4	^[OS
F6	F6	^[[17~
F7	F7	^[[18~
F8	F8	^[[19~
F9	F9	^[[20~
F10	F10	^[[21~
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Insert Here	^[[2~
Gray Delete	Remove	^[[3~
Gray Home	Find	^[[1~
Gray End	Select	^[[4~
Gray Pageup	Prev Screen	^[[5~
Gray Pagedown	Next Screen	^[[6~
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Numlock	PF1	^[OP
Pad /	PF2	^[OQ
Pad *	PF3	^[OR
Pad -	PF4	^[OS
Pad +	Keypad MINUS	-
Pad Enter	Enter	^M
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5

Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+F1	F11	^[[23~
Shift+F2	F12	^[[24~
Shift+F3	F13	^[[25~
Shift+F4	F14	^[[26~
Shift+F5	Help	^[[28~
Shift+F6	Do	^[[29~
Shift+F7	F17	^[[31~
Shift+F8	F18	^[[32~
Shift+F9	F19	^[[33~
Shift+F10	F20	^[[34~
Shift+Gray Delete	Delete	Δ
Shift+Gray Home	Home Cursor	^[[H
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Delete	Delete	Δ
Shift+Pad End	Erase End of Line	^[[K
Shift+Pad Down	Cursor Down	^[[B
Shift+Pad Pagedown	Clear Screen	^[[H^[[2J
Shift+Pad Left	Cursor Left	^[[D
Shift+Pad Right	Cursor Right	^[[C
Shift+Pad Home	Home Cursor	^[[H
Shift+Pad Up	Cursor Up	^[[A
Shift+Pad Pageup	Delete Line	^[[M
Control+F6	Shifted F6	^[[37~
Control+F7	Shifted F7	^[[38~
Control+F8	Shifted F8	^[[39~
Control+F9	Shifted F9	^[[40~
Control+F10	Shifted F10	^[[41~
Control+Backspace	Delete	Δ
Control+Gray Insert	Insert Line	^[[L
Control+Gray Delete	Delete Line	^[[M
Control+Gray End	Erase End of Line	^[[K
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Gray Up	Application Cursor Up	^[OA
Control+Gray Down	Application Cursor Down	^[OB
Control+Gray Left	Application Cursor Left	^[OD
Control+Gray Right	Application Cursor Right	^[OC
Control+Pad /	Keypad Application 00	^[Oo
Control+Pad -	Keypad Application MINUS	^[Om

Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad .	Keypad Application PERIOD	^[On
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application PERIOD	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 5	^[Ot
Control+Pad Clear	Keypad Application 6	^[Ou
Control+Pad Right	Keypad Application 7	^[Ov
Control+Pad Home	Keypad Application 8	^[Ow
Control+Pad Up	Keypad Application 9	^[Ox
Control+Pad Pageup	Keypad Application 0	^[Oy
Shift+Control+F1	Shifted F6	^[[43~
Shift+Control+F2	Shifted F7	^[[44~
Shift+Control+F3	Shifted F8	^[[45~
Shift+Control+F4	Shifted F9	^[[46~
Shift+Control+F5	Shifted Help	^[[48~
Shift+Control+F6	Shifted Do	^[[49~
Shift+Control+F7	Shifted F17	^[[51~
Shift+Control+F8	Shifted F18	^[[52~
Shift+Control+F9	Shifted F19	^[[53~
Shift+Control+F10	Shifted F20	^[[54~
Shift+Control+Pad Down	Application Cursor Down	^[OB
Shift+Control+Pad Left	Application Cursor Left	^[OD
Shift+Control+Pad Right	Application Cursor Right	^[OC
Shift+Control+Pad Home	Insert Line	^[[L
Shift+Control+Pad Up	Application Cursor Up	^[OA

TVI 925

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M
F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Function 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M
F8	Function 8 (F8)	^AG^M
F9	Function 9 (F9)	^AH^M
F10	Function 10 (F10)	^AI^M
Backspace	Backspace	^H

Enter	Enter	^M
Gray Insert	Character Insert	^[Q
Gray Delete	Character Delete	^[W
Gray Home	Home Cursor	^^
Gray End	Erase End of Line	^[T
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^V
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Character Delete	^[W
Pad Insert	Character Insert	^[Q
Pad End	Erase End of Line	^[T
Pad Down	Cursor Down	^V
Pad Left	Cursor Left	^H
Pad Right	Cursor Right	^L
Pad Home	Home Cursor	^^
Pad Up	Cursor Up	^K
Shift+Tab	Back Tab	^[I
Shift+F1	Function 12 (F12)	^A`^M
Shift+F2	Function 13 (F13)	^Aa^M
Shift+F3	Function 14 (F14)	^Ab^M
Shift+F4	Function 15 (F15)	^Ac^M
Shift+F5	Function 16 (F16)	^Ad^M
Shift+F6	Function 17 (F17)	^Ae^M
Shift+F7	Function 18 (F18)	^Af^M
Shift+F8	Function 19 (F19)	^Ag^M
Shift+F9	Function 20(F20)	^Ah^M
Shift+F10	Function 21 (F21)	^Ai^M
Shift+Gray Delete	Line Delete	^[R
Shift+Gray Home	Clear Page	^[{
Shift+Gray End	Erase End of Page	^[Y
Shift+Gray Pageup	Line Insert	^[E
Shift+Gray Up	Shift Cursor Up	^[j
Shift+Gray Down	Shift Cursor Down	^J
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.

Shift+Pad Insert	0	0
Shift+Pad End		^[Y
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	Clear Page	^[{
Shift+Pad Up	8	8
Shift+Pad Pageup	9	9
Control+F7	Character Insert	^[Q
Control+F8	Character Delete	^[W
Control+F9	Line Insert	^[E
Control+F10	Line Delete	^[R
Control+Backspace	Delete Character	Δ
Control+Gray Delete	Shift Character Delete	^[r
Control+Gray Home	Clear All	^Z
Control+Pad Delete	Line Delete	^[R
Control+Pad Insert	Line Insert	^[E
Control+Pad Home	Clear All	^Z
Shift+Control+F7	Shift Character Insert	^[q
Shift+Control+F8	Shift Character Delete	^[r
Shift+Control+F9	Shift Line Insert	^[N
Shift+Control+F10	Shift Line Delete	^[O
Shift+Control+Pad End	1	1
Shift+Control+Pad Home	7	7

TVI 950

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M
F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Function 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M
F8	Function 8 (F8)	^AG^M
F9	Function 9 (F9)	^AH^M
F10	Function 10 (F10)	^AI^M
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Character Insert	^[Q
Gray Delete	Character Delete	^[W
Gray Home	Home Cursor	^^
Gray End	Erase End of Line	^[T
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^V
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+

Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Character Delete	^[W
Pad Insert	Character Insert	^[Q
Pad End	Erase End of Line	^[T
Pad Down	Cursor Down	^V
Pad Left	Cursor Left	^H
Pad Right	Cursor Right	^L
Pad Home	Home Cursor	^^
Pad Up	Cursor Up	^K
Shift+Tab	Back Tab	^[
Shift+F1	Function 12 (F12)	^A`^M
Shift+F2	Function 13 (F13)	^Aa^M
Shift+F3	Function 14 (F14)	^Ab^M
Shift+F4	Function 15 (F15)	^Ac^M
Shift+F5	Function 16 (F16)	^Ad^M
Shift+F6	Function 17 (F17)	^Ae^M
Shift+F7	Function 18 (F18)	^Af^M
Shift+F8	Function 19 (F19)	^Ag^M
Shift+F9	Function 20(F20)	^Ah^M
Shift+F10	Function 21 (F21)	^Ai^M
Shift+Gray Delete	Line Delete	^[R
Shift+Gray Home	Clear Page	^[{
Shift+Gray End	Erase End of Page	^[Y
Shift+Gray Pageup	Line Insert	^[E
Shift+Gray Up	Shift Cursor Up	^j
Shift+Gray Down	Shift Cursor Down	^J
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End		^[Y
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	Clear Page	^[{
Shift+Pad Up	8	8
Shift+Pad Pageup	9	9
Control+F7	Character Insert	^[Q
Control+F8	Character Delete	^[W
Control+F9	Line Insert	^[E

QmodemPro Key	Terminal Key	Characters Sent
Control+F10	Line Delete	^[R
Control+Backspace	Delete Character	Δ
Control+Gray Delete	Shift Character Delete	^[r
Control+Gray Home	Clear All	^Z
Control+Pad Delete	Line Delete	^[R
Control+Pad Insert	Line Insert	^[E
Control+Pad Home	Clear All	^Z
Shift+Control+F7	Shift Character Insert	^[q
Shift+Control+F8	Shift Character Delete	^[r
Shift+Control+F9	Shift Line Insert	^[N
Shift+Control+F10	Shift Line Delete	^[O
Shift+Control+Pad End	1	1
Shift+Control+Pad Home	7	7

TVI 955

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M
F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Function 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M
F8	Function 8 (F8)	^AG^M
F9	Function 9 (F9)	^AH^M
F10	Function 10 (F10)	^AI^M
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Character Insert	^[Q
Gray Delete	Character Delete	^[W
Gray Home	Home Cursor	^^
Gray End	Erase End of Line	^[T
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^V
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Character Delete	^[W

Pad Insert	Character Insert	^[Q
Pad End	Erase End of Line	^[T
Pad Down	Cursor Down	^V
Pad Left	Cursor Left	^H
Pad Right	Cursor Right	^L
Pad Home	Home Cursor	^^
Pad Up	Cursor Up	^K
Shift+Tab	Back Tab	^[I
Shift+F1	Function 12 (F12)	^A`^M
Shift+F2	Function 13 (F13)	^Aa^M
Shift+F3	Function 14 (F14)	^Ab^M
Shift+F4	Function 15 (F15)	^Ac^M
Shift+F5	Function 16 (F16)	^Ad^M
Shift+F6	Function 17 (F17)	^Ae^M
Shift+F7	Function 18 (F18)	^Af^M
Shift+F8	Function 19 (F19)	^Ag^M
Shift+F9	Function 20(F20)	^Ah^M
Shift+F10	Function 21 (F21)	^Ai^M
Shift+Gray Delete	Line Delete	^[R
Shift+Gray Home	Clear Page	^[{
Shift+Gray End	Erase End of Page	^[Y
Shift+Gray Pageup	Line Insert	^[E
Shift+Gray Up	Shift Cursor Up	^j
Shift+Gray Down	Shift Cursor Down	^J
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	.
Shift+Pad Insert	0	0
Shift+Pad End		^[Y
Shift+Pad Down	2	2
Shift+Pad Pagedown	3	3
Shift+Pad Left	4	4
Shift+Pad Clear	5	5
Shift+Pad Right	6	6
Shift+Pad Home	Clear Page	^[{
Shift+Pad Up	8	8
Shift+Pad Pageup	9	9
Control+F7	Character Insert	^[Q
Control+F8	Character Delete	^[W
Control+F9	Line Insert	^[E
Control+F10	Line Delete	^[R
Control+Backspace	Delete Character	Δ
Control+Gray Delete	Shift Character Delete	^[r
Control+Gray Home	Clear All	^Z
Control+Pad Delete	Line Delete	^[R
Control+Pad Insert	Line Insert	^[E
Control+Pad Home	Clear All	^Z
Shift+Control+F7	Shift Character Insert	^[q
Shift+Control+F8	Shift Character Delete	^[r
Shift+Control+F9	Shift Line Insert	^[N
Shift+Control+F10	Shift Line Delete	^[O
Shift+Control+Pad End	1	1
Shift+Control+Pad Home	7	7

Vidtex

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Program Function 1 (PF1)	^[P
F2	Program Function 2 (PF2)	^[Q
F3	Program Function 3 (PF3)	^[R
F4	Program Function 4 (PF4)	^[S
Backspace	Horizontal Backspace	^H
Enter	Enter	^M
Gray Delete	Character Delete	Δ
Gray Home	Home Cursor	^[H
Gray End	Erase End of Line	^[K
Gray Up	Cursor Up	^[A
Gray Down	Cursor Down	^[B
Gray Left	Cursor Left	^[D
Gray Right	Cursor Right	^[C
Numlock	Program Function 1 (PF1)	^[P
Pad /	Program Function 2 (PF2)	^[Q
Pad *	Program Function 3 (PF3)	^[R
Pad -	Program Function 4 (PF4)	^[S
Pad +	Keypad COMMA	,
Pad Enter	Enter	^M
Pad .	Keypad PERIOD	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Keypad PERIOD	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+Enter	Enter	^M
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	Keypad MINUS	-
Shift+Pad +	Keypad MINUS	-
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Character Delete	Δ
Shift+Pad End	Delete End of Line	^[K
Shift+Pad Down	Cursor Down	^[B

QmodemPro Key	Terminal Key	Characters Sent
Shift+Pad Pagedown	Clear Screen	^[H^[J
Shift+Pad Left	Cursor Left	^[D
Shift+Pad Right	Cursor Right	^[C
Shift+Pad Home	Home Cursor	^[H
Shift+Pad Up	Cursor Up	^[A
Shift+Pad Pageup	Delete Line	^[M
Control+Backspace	Character Delete	Δ
Control+Enter	Enter	^M
Control+Gray End	Erase End of Page	^[J
Control+Gray Pageup	Delete Line	^[M
Control+Gray Pagedown	Clear Screen	^[H^[J
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	Keypad Application MINUS	^[?m
Control+Pad +	Keypad Application COMMA	^[?l
Control+Pad Enter	Keypad Application Enter	^[?M
Control+Pad .	Keypad Application PERIOD	^[?n
Control+Pad 0	Keypad Application 0	^[?p
Control+Pad 1	Keypad Application 1	^[?q
Control+Pad 2	Keypad Application 2	^[?r
Control+Pad 3	Keypad Application 3	^[?s
Control+Pad 4	Keypad Application 4	^[?t
Control+Pad 5	Keypad Application 5	^[?u
Control+Pad 6	Keypad Application 6	^[?v
Control+Pad 7	Keypad Application 7	^[?w
Control+Pad 8	Keypad Application 8	^[?x
Control+Pad 9	Keypad Application 9	^[?y
Control+Pad Delete	Keypad Application PERIOD	^[?n
Control+Pad Insert	Keypad Application 0	^[?p
Control+Pad End	Keypad Application 1	^[?q
Control+Pad Down	Keypad Application 2	^[?r
Control+Pad Pagedown	Keypad Application 3	^[?s
Control+Pad Left	Keypad Application 4	^[?t
Control+Pad Clear	Keypad Application 5	^[?u
Control+Pad Right	Keypad Application 6	^[?v
Control+Pad Home	Keypad Application 7	^[?w
Control+Pad Up	Keypad Application 8	^[?x
Control+Pad Pageup	Keypad Application 9	^[?y
Shift+Control+Enter	Enter	^M
Shift+Control+Pad Enter	Enter	^M

VT 52

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Program Function 1 (PF1)	^[P
F2	Program Function 2 (PF2)	^[Q
F3	Program Function 3 (PF3)	^[R
F4	Program Function 4 (PF4)	^[S
Backspace	Horizontal Backspace	^H
Enter	Enter	^M
Gray Delete	Character Delete	Δ
Gray Home	Home Cursor	^[H
Gray End	Erase End of Line	^[K
Gray Up	Cursor Up	^[A

Gray Down	Cursor Down	^[B
Gray Left	Cursor Left	^[D
Gray Right	Cursor Right	^[C
Numlock	Program Function 1 (PF1)	^[P
Pad /	Program Function 2 (PF2)	^[Q
Pad *	Program Function 3 (PF3)	^[R
Pad -	Program Function 4 (PF4)	^[S
Pad +	Keypad COMMA	,
Pad Enter	Enter	^M
Pad .	Keypad PERIOD	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Keypad PERIOD	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+Enter	Enter	^M
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	Keypad MINUS	-
Shift+Pad +	Keypad MINUS	-
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Character Delete	Δ
Shift+Pad End	Delete End of Line	^[K
Shift+Pad Down	Cursor Down	^[B
Shift+Pad Pagedown	Clear Screen	^[H^[J
Shift+Pad Left	Cursor Left	^[D
Shift+Pad Right	Cursor Right	^[C
Shift+Pad Home	Home Cursor	^[H
Shift+Pad Up	Cursor Up	^[A
Shift+Pad Pageup	Delete Line	^[M
Control+Backspace	Character Delete	Δ
Control+Enter	Enter	^M
Control+Gray End	Erase End of Page	^[J
Control+Gray Pageup	Delete Line	^[M
Control+Gray Pagedown	Clear Screen	^[H^[J
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	Keypad Application MINUS	^[?m
Control+Pad +	Keypad Application COMMA	^[?l
Control+Pad Enter	Keypad Application Enter	^[?M

Control+Pad .	Keypad Application PERIOD	^[?n
Control+Pad 0	Keypad Application 0	^[?p
Control+Pad 1	Keypad Application 1	^[?q
Control+Pad 2	Keypad Application 2	^[?r
Control+Pad 3	Keypad Application 3	^[?s
Control+Pad 4	Keypad Application 4	^[?t
Control+Pad 5	Keypad Application 5	^[?u
Control+Pad 6	Keypad Application 6	^[?v
Control+Pad 7	Keypad Application 7	^[?w
Control+Pad 8	Keypad Application 8	^[?x
Control+Pad 9	Keypad Application 9	^[?y
Control+Pad Delete	Keypad Application PERIOD	^[?n
Control+Pad Insert	Keypad Application 0	^[?p
Control+Pad End	Keypad Application 1	^[?q
Control+Pad Down	Keypad Application 2	^[?r
Control+Pad Pagedown	Keypad Application 3	^[?s
Control+Pad Left	Keypad Application 4	^[?t
Control+Pad Clear	Keypad Application 5	^[?u
Control+Pad Right	Keypad Application 6	^[?v
Control+Pad Home	Keypad Application 7	^[?w
Control+Pad Up	Keypad Application 8	^[?x
Control+Pad Pageup	Keypad Application 9	^[?y
Shift+Control+Enter	Enter	^M
Shift+Control+Pad Enter	Enter	^M

VT 100

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Program Function 1 (PF1)	^[OP
F2	Program Function 2 (PF2)	^[OQ
F3	Program Function 3 (PF3)	^[OR
F4	Program Function 4 (PF4)	^[OS
Backspace	Backspace	^H
Enter	Return	^M
Gray Delete	Delete Character	Δ
Gray Home	Home Cursor	^[[H
Gray End	Erase End of Line	^[[K
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Numlock	PF1	^[OP
Pad /	PF2	^[OQ
Pad *	PF3	^[OR
Pad -	PF4	^[OS
Pad +	Keypad COMMA	,
Pad Enter	Enter	^M
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6

Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+Backspace	Delete Character	Δ
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete Character	Δ
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	Keypad MINUS	-
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	Δ
Shift+Pad End	Erase End of Line	^[[K
Shift+Pad Down	Cursor Down	^[[B
Shift+Pad Pagedown	Clear Screen	^[[H^[[2J
Shift+Pad Left	Cursor Left	^[[D
Shift+Pad Right	Cursor Right	^[[C
Shift+Pad Home	Home Cursor	^[[H
Shift+Pad Up	Cursor Up	^[[A
Shift+Pad Pageup	Delete Line	^[[M
Control+Backspace	Ctrl-Backspace	^X
Control+Enter	Enter	^M
Control+Gray Home	Insert Line	^[[L
Control+Gray Pageup	Delete Line	^[[M
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Gray Up	Application Cursor Up	^[OA
Control+Gray Down	Application Cursor Down	^[OB
Control+Gray Left	Application Cursor Left	^[OD
Control+Gray Right	Application Cursor Right	^[OC
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad .	Keypad Application PERIOD	^[On
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy

Control+Pad Delete	Keypad Application PERIOD	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot
Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 9	^[Oy
Shift+Control+Enter	Enter	^M
Shift+Control+Pad Enter	Enter	^M
Shift+Control+Pad Down	Application Cursor Down	^[OB
Shift+Control+Pad Left	Application Cursor Left	^[OD
Shift+Control+Pad Right	Application Cursor Right	^[OC
Shift+Control+Pad Home	Insert LIne	^[[L
Shift+Control+Pad Up	Application Cursor Up	^[OA

VT 102

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Program Function 1 (PF1)	^[OP
F2	Program Function 2 (PF2)	^[OQ
F3	Program Function 3 (PF3)	^[OR
F4	Program Function 4 (PF4)	^[OS
Backspace	Backspace	^H
Enter	Return	^M
Gray Delete	Delete Character	Δ
Gray Home	Home Cursor	^[[H
Gray End	Erase End of Line	^[[K
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Numlock	PF1	^[OP
Pad /	PF2	^[OQ
Pad *	PF3	^[OR
Pad -	PF4	^[OS
Pad +	Keypad COMMA	,
Pad Enter	Enter	^M
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1

Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+Backspace	Delete Character	Δ
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete Character	Δ
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	Keypad MINUS	-
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	.	Δ
Shift+Pad End	Erase End of Line	^[[K
Shift+Pad Down	Cursor Down	^[[B
Shift+Pad Pagedown	Clear Screen	^[[H^[[2J
Shift+Pad Left	Cursor Left	^[[D
Shift+Pad Right	Cursor Right	^[[C
Shift+Pad Home	Home Cursor	^[[H
Shift+Pad Up	Cursor Up	^[[A
Shift+Pad Pageup	Delete Line	^[[M
Control+Backspace	Ctrl-Backspace	^X
Control+Enter	Enter	^M
Control+Gray Home	Insert Line	^[[L
Control+Gray Pageup	Delete Line	^[[M
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Gray Up	Application Cursor Up	^[OA
Control+Gray Down	Application Cursor Down	^[OB
Control+Gray Left	Application Cursor Left	^[OD
Control+Gray Right	Application Cursor Right	^[OC
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad .	Keypad Application PERIOD	^[On
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application PERIOD	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot

Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 9	^[Oy
Shift+Control+Enter	Enter	^M
Shift+Control+Pad Enter	Enter	^M
Shift+Control+Pad Down	Application Cursor Down	^[OB
Shift+Control+Pad Left	Application Cursor Left	^[OD
Shift+Control+Pad Right	Application Cursor Right	^[OC
Shift+Control+Pad Home	Insert LIne	^[[L
Shift+Control+Pad Up	Application Cursor Up	^[OA

VT 220

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	PF1	^[OP
F2	PF2	^[OQ
F3	PF3	^[OR
F4	PF4	^[OS
F5	Next Screen	^[[6~
F6	F6	^[[17~
F7	F7	^[[18~
F8	F8	^[[19~
F9	F9	^[[20~
F10	F10	^[[21~
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Insert Here	^[[2~
Gray Delete	Remove	^[[3~
Gray Home	Find	^[[1~
Gray End	Select	^[[4~
Gray Pageup	Prev Screen	^[[5~
Gray Pagedown	Next Screen	^[[6~
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Numlock	PF1	^[OP
Pad /	PF2	^[OQ
Pad *	PF3	^[OR
Pad -	PF4	^[OS
Pad +	Keypad MINUS	-
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8

Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+F1	F11	^[[23~
Shift+F2	F12	^[[24~
Shift+F3	F13	^[[25~
Shift+F4	F14	^[[26~
Shift+F5	F15	^[[28~
Shift+F6	F16	^[[29~
Shift+F7	F17	^[[31~
Shift+F8	F18	^[[32~
Shift+F9	F19	^[[33~
Shift+F10	F20	^[[34~
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete	Δ
Shift+Gray Home	Home Cursor	^[[H
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Delete	Δ
Shift+Pad End	Erase End of Line	^[[K
Shift+Pad Down	Cursor Down	^[[B
Shift+Pad Pagedown	Clear Screen	^[[H^[[2J
Shift+Pad Left	Cursor Left	^[[D
Shift+Pad Right	Cursor Right	^[[C
Shift+Pad Home	Home Cursor	^[[H
Shift+Pad Up	Cursor Up	^[[A
Shift+Pad Pageup	Delete Line	^[[M
Control+F1	Find	^[[1~
Control+F2	Insert Here	^[[2~
Control+F3	Remove	^[[3~
Control+F4	Select	^[[4~
Control+F5	Prev Screen	^[[5~
Control+Backspace	Delete	Δ
Control+Enter	Enter	^M
Control+Gray Insert	Insert Line	^[[L
Control+Gray Delete	Delete Line	^[[M
Control+Gray End	Erase End of Line	^[[K
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Gray Up	Application Cursor Up	^[OA
Control+Gray Down	Application Cursor Down	^[OB
Control+Gray Left	Appplication Cursor Left	^[OD
Control+Gray Right	Application Cursor Right	^[OC
Control+Pad /	/	/
Control+Pad *	*	*

Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad .	Keypad Application .	^[On
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application .	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot
Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 8	^[Oy
Shift+Control+Pad Down	Application Cursor Down	^[OB
Shift+Control+Pad Left	Application Cursor Left	^[OD
Shift+Control+Pad Right	Application Cursor Right	^[OC
Shift+Control+Pad Home	Insert Line	^[[L
Shift+Control+Pad Up	Application Cursor Up	^[OA

VT 320

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	PF1	^[OP
F2	PF2	^[OQ
F3	PF3	^[OR
F4	PF4	^[OS
F5	Next Screen	^[[6~
F6	F6	^[[17~
F7	F7	^[[18~
F8	F8	^[[19~
F9	F9	^[[20~
F10	F10	^[[21~
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Insert Here	^[[2~
Gray Delete	Remove	^[[3~
Gray Home	Find	^[[1~
Gray End	Select	^[[4~
Gray Pageup	Prev Screen	^[[5~
Gray Pagedown	Next Screen	^[[6~
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B

Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Numlock	PF1	^[OP
Pad /	PF2	^[OQ
Pad *	PF3	^[OR
Pad -	PF4	^[OS
Pad +	Keypad MINUS	-
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+F1	F11	^[[23~
Shift+F2	F12	^[[24~
Shift+F3	F13	^[[25~
Shift+F4	F14	^[[26~
Shift+F5	F15	^[[28~
Shift+F6	F16	^[[29~
Shift+F7	F17	^[[31~
Shift+F8	F18	^[[32~
Shift+F9	F19	^[[33~
Shift+F10	F20	^[[34~
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete	Δ
Shift+Gray Home	Home Cursor	^[[H
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Delete	Δ
Shift+Pad End	Erase End of Line	^[[K
Shift+Pad Down	Cursor Down	^[[B
Shift+Pad Pagedown	Clear Screen	^[[H^[[2J
Shift+Pad Left	Cursor Left	^[[D
Shift+Pad Right	Cursor Right	^[[C
Shift+Pad Home	Home Cursor	^[[H
Shift+Pad Up	Cursor Up	^[[A

QmodemPro Key	Terminal Key	Characters Sent
Shift+Pad Pageup	Delete Line	^[[M
Control+F1	Find	^[[1~
Control+F2	Insert Here	^[[2~
Control+F3	Remove	^[[3~
Control+F4	Select	^[[4~
Control+F5	Prev Screen	^[[5~
Control+Backspace	Delete	Δ
Control+Enter	Enter	^M
Control+Gray Insert	Insert Line	^[[L
Control+Gray Delete	Delete Line	^[[M
Control+Gray End	Erase End of Line	^[[K
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Gray Up	Application Cursor Up	^[OA
Control+Gray Down	Application Cursor Down	^[OB
Control+Gray Left	Appplication Cursor Left	^[OD
Control+Gray Right	Application Cursor Right	^[OC
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad .	Keypad Application .	^[On
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application .	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot
Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 8	^[Oy
Shift+Control+Pad Down	Application Cursor Down	^[OB
Shift+Control+Pad Left	Application Cursor Left	^[OD
Shift+Control+Pad Right	Application Cursor Right	^[OC
Shift+Control+Pad Home	Insert Line	^[[L
Shift+Control+Pad Up	Application Cursor Up	^[OA

Wyse 30

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M

F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Funciton 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M
F8	Function 8 (F8)	^AG^M
F9	Function 9 (F9)	^AH^M
F10	Function 10 (F10)	^AI^M
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Insert Character	^[Q
Gray Delete	Delete Character	^[W
Gray Home	Home Cursor	^^
Gray End	Clear Line	^[T
Gray Pageup	Page Prev	^[K
Gray Pagedown	Page Next	^[J
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^J
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Numlock	Insert Character	^[Q
Pad /	Delete Character	^[W
Pad *	Clear Line	^[T
Pad -	Replace	^[r
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	7	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+Tab	Back Tab	^[I
Shift+F1	Function 17 (Shift F1)	^A`^M
Shift+F2	Function 18 (Shift F2)	^Aa^M
Shift+F3	Function 19 (Shift F3)	^Ab^M
Shift+F4	Function 20 (Shift F4)	^Ac^M
Shift+F5	Function 21 (Shift F5)	^Ad^M
Shift+F6	Function 22 (Shift F6)	^Ae^M
Shift+F7	Function 23 (Shift F7)	^Af^M

Shift+F8	Function 24 (Shift F8)	^Ag^M
Shift+F9	Function 25 (Shift F9)	^Ah^M
Shift+F10	Function 26 (Shift F10)	^Ai^M
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete Line	^[R
Shift+Gray Home	Shift Home Cursor	^[{
Shift+Gray End	Erase End of Page	^[Y
Shift+Gray Pageup	Insert Line	^[E
Shift+Numlock	Insert Line	^[E
Shift+Pad /	Delete Line	^[R
Shift+Pad *	Clear Screen	^[Y
Shift+Pad -	Insert	^[q
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Delete Character	^[W
Shift+Pad Insert	Insert Character	^[Q
Shift+Pad End	Clear Line	^[T
Shift+Pad Down	Cursor Down	^J
Shift+Pad Pagedown	Page Next	^[J
Shift+Pad Left	Backspace	^H
Shift+Pad Clear	5	5
Shift+Pad Right	Cursor Right	^L
Shift+Pad Home	Home Cursor	^^
Shift+Pad Up	Cursor Up	^K
Shift+Pad Pageup	Page Prev	^[K
Control+F1	Function 13 (F13)	^AL^M
Control+F2	Function 14 (F14)	^AM^M
Control+F3	Function 15 (F15)	^AN^M
Control+F4	Function 16 (F16)	^AO^M
Control+F5	Function 29 (Shift F13)	^Al^M
Control+F6	Function 30 (Shift F14)	^Am^M
Control+F7	Function 31 (Shift F15)	^An^M
Control+F8	Function 32 (Shift F16)	^Ao^M
Control+Backspace	Ctrl-Backspace	Δ

Wyse 50

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M
F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Funciton 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M
F8	Function 8 (F8)	^AG^M
F9	Function 9 (F9)	^AH^M
F10	Function 10 (F10)	^AI^M
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Insert Character	^[Q
Gray Delete	Delete Character	^[W
Gray Home	Home Cursor	^^
Gray End	Clear Line	^[T
Gray Pageup	Page Prev	^[K
Gray Pagedown	Page Next	^[J
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^J
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Numlock	Insert Character	^[Q
Pad /	Delete Character	^[W
Pad *	Clear Line	^[T
Pad -	Replace	^[r
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	7	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9

Shift+Tab	Back Tab	^[I
Shift+F1	Function 17 (Shift F1)	^A`^M
Shift+F2	Function 18 (Shift F2)	^Aa^M
Shift+F3	Function 19 (Shift F3)	^Ab^M
Shift+F4	Function 20 (Shift F4)	^Ac^M
Shift+F5	Function 21 (Shift F5)	^Ad^M
Shift+F6	Function 22 (Shift F6)	^Ae^M
Shift+F7	Function 23 (Shift F7)	^Af^M
Shift+F8	Function 24 (Shift F8)	^Ag^M
Shift+F9	Function 25 (Shift F9)	^Ah^M
Shift+F10	Function 26 (Shift F10)	^Ai^M
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete Line	^[R
Shift+Gray Home	Shift Home Cursor	^[{
Shift+Gray End	Erase End of Page	^[Y
Shift+Gray Pageup	Insert Line	^[E
Shift+Numlock	Insert Line	^[E
Shift+Pad /	Delete Line	^[R
Shift+Pad *	Clear Screen	^[Y
Shift+Pad -	Insert	^[q
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Delete Character	^[W
Shift+Pad Insert	Insert Character	^[Q
Shift+Pad End	Clear Line	^[T
Shift+Pad Down	Cursor Down	^J
Shift+Pad Pagedown	Page Next	^[J
Shift+Pad Left	Backspace	^H
Shift+Pad Clear	5	5
Shift+Pad Right	Cursor Right	^L
Shift+Pad Home	Home Cursor	^^
Shift+Pad Up	Cursor Up	^K
Shift+Pad Pageup	Page Prev	^[K
Control+F1	Function 13 (F13)	^AL^M
Control+F2	Function 14 (F14)	^AM^M
Control+F3	Function 15 (F15)	^AN^M
Control+F4	Function 16 (F16)	^AO^M
Control+F5	Function 29 (Shift F13)	^AI^M
Control+F6	Function 30 (Shift F14)	^Am^M
Control+F7	Function 31 (Shift F15)	^An^M
Control+F8	Function 32 (Shift F16)	^Ao^M
Control+Backspace	Ctrl-Backspace	Δ

Wyse 60

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M
F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Funciton 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M
F8	Function 8 (F8)	^AG^M
F9	Function 9 (F9)	^AH^M
F10	Function 10 (F10)	^AI^M
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Insert Character	^[Q
Gray Delete	Delete Character	^[W
Gray Home	Home Cursor	^^
Gray End	Clear Line	^[T
Gray Pageup	Page Prev	^[K
Gray Pagedown	Page Next	^[J
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^J
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Numlock	Insert Character	^[Q
Pad /	Delete Character	^[W
Pad *	Clear Line	^[T
Pad -	Replace	^[r
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	7	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9

Shift+Tab	Back Tab	^[I
Shift+F1	Function 17 (Shift F1)	^A`^M
Shift+F2	Function 18 (Shift F2)	^Aa^M
Shift+F3	Function 19 (Shift F3)	^Ab^M
Shift+F4	Function 20 (Shift F4)	^Ac^M
Shift+F5	Function 21 (Shift F5)	^Ad^M
Shift+F6	Function 22 (Shift F6)	^Ae^M
Shift+F7	Function 23 (Shift F7)	^Af^M
Shift+F8	Function 24 (Shift F8)	^Ag^M
Shift+F9	Function 25 (Shift F9)	^Ah^M
Shift+F10	Function 26 (Shift F10)	^Ai^M
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete Line	^[R
Shift+Gray Home	Shift Home Cursor	^[{
Shift+Gray End	Erase End of Page	^[Y
Shift+Gray Pageup	Insert Line	^[E
Shift+Numlock	Insert Line	^[E
Shift+Pad /	Delete Line	^[R
Shift+Pad *	Clear Screen	^[Y
Shift+Pad -	Insert	^[q
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Delete Character	^[W
Shift+Pad Insert	Insert Character	^[Q
Shift+Pad End	Clear Line	^[T
Shift+Pad Down	Cursor Down	^J
Shift+Pad Pagedown	Page Next	^[J
Shift+Pad Left	Backspace	^H
Shift+Pad Clear	5	5
Shift+Pad Right	Cursor Right	^L
Shift+Pad Home	Home Cursor	^^
Shift+Pad Up	Cursor Up	^K
Shift+Pad Pageup	Page Prev	^[K
Control+F1	Function 13 (F13)	^AL^M
Control+F2	Function 14 (F14)	^AM^M
Control+F3	Function 15 (F15)	^AN^M
Control+F4	Function 16 (F16)	^AO^M
Control+F5	Function 29 (Shift F13)	^AI^M
Control+F6	Function 30 (Shift F14)	^Am^M
Control+F7	Function 31 (Shift F15)	^An^M
Control+F8	Function 32 (Shift F16)	^Ao^M
Control+Backspace	Ctrl-Backspace	Δ

Wyse 75

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Print Copy Key (F1)	^[[?5i
F2	Print Trans Key (F2)	^[[?3i
F3	Screen Send (F3)	^[[2i
F4	Insert Char	^[[@
F5	Line Delete	^[[M
F6	F6	^[[17~
F7	F7	^[[18~
F8	F8	^[[19~

F9	F9	^[[20~
F10	F10	^[[21~
Backspace	Backspace	^H
Enter	Return	^M
Gray Insert	Shift F12 (Insert Here)	^[[2~
Gray Delete	Shift F13 (Remove)	^[[3~
Gray Home	Shift F11 (Find)	^[[1~
Gray End	Shift F14 (Select)	^[[4~
Gray Pageup	Shift F15 (Prev Screen)	^[[5~
Gray Pagedown	Shift F16 (Next Screen)	^[[6~
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Numlock	PF1	^[OP
Pad /	PF2	^[OQ
Pad *	PF3	^[OR
Pad -	PF4	^[OS
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+F1	F11	^[[23~
Shift+F2	F12	^[[24~
Shift+F3	F13	^[[25~
Shift+F4	F14	^[[26~
Shift+F5	F15	^[[28~
Shift+F6	F16	^[[29~
Shift+Enter	Enter	^M
Shift+Gray Delete	Character Delete	Δ
Shift+Gray Home	Home Cursor	^[[H
Shift+Pad +	Keypad MINUS	-
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Character Delete	Δ
Shift+Pad End	Line Clear	^[[K
Shift+Pad Down	Cursor Down	^[[B

Shift+Pad Pagedown	Clear Screen	^[[H^[[2J
Shift+Pad Left	Cursor Left	^[[D
Shift+Pad Right	Cursor Right	^[[C
Shift+Pad Home	Home Cursor	^[[H
Shift+Pad Up	Cursor Up	^[[A
Shift+Pad Pageup	Delete Line	^[[M
Control+F1	Shift F1	^[[4i
Control+F2	Shift F2	^[[?1i
Control+F3	Shift F3	^[[0i
Control+F4	Insert Line	^[[L
Control+F5	Line Clear	^[[K
Control+F6	Shift F6	^[[31~
Control+F7	Shift F7	^[[32~
Control+F8	Shift F8	^[[33~
Control+F9	Shift F9	^[[34~
Control+F10	Shift F10	^[[35~
Control+Backspace	Character Delete	Δ
Control+Enter	Enter	^M
Control+Gray Home	Insert Line	^[[L
Control+Gray End	Erase End of Line	^[[K
Control+Gray Pageup	Delete Line	^[[M
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Gray Up	Application Cursor Up	^[OA
Control+Gray Down	Application Cursor Down	^[OB
Control+Gray Left	Application Cursor Left	^[OD
Control+Gray Right	Application Cursor Right	^[OC
Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application .	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot
Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 9	^[Oy
Shift+Control+F1	Find	^[[1~
Shift+Control+F2	Select	^[[2~
Shift+Control+F3	Insert Here	^[[3~
Shift+Control+F4	Remove	^[[4~
Shift+Control+F5	Prev Screen	^[[5~
Shift+Control+F6	Next Screen	^[[6~

Shift+Control+Pad Down	Application Cursor Down	^[OB
Shift+Control+Pad Left	Application Cursor Left	^[OD
Shift+Control+Pad Right	Application Cursor Right	^[OC
Shift+Control+Pad Home	Insert Line	^[[L
Shift+Control+Pad Up	Application Cursor Up4	^[OA

Wyse 85

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Print Copy Key (F1)	^[[?5i
F2	Print Trans Key (F2)	^[[?3i
F3	Screen Send (F3)	^[[2i
F4	Insert Char	^[[@
F5	Line Delete	^[[M
F6	F6	^[[17~
F7	F7	^[[18~
F8	F8	^[[19~
F9	F9	^[[20~
F10	F10	^[[21~
Backspace	Backspace	^H
Enter	Return	^M
Gray Insert	Shift F12 (Insert Here)	^[[2~
Gray Delete	Shift F13 (Remove)	^[[3~
Gray Home	Shift F11 (Find)	^[[1~
Gray End	Shift F14 (Select)	^[[4~
Gray Pageup	Shift F15 (Prev Screen)	^[[5~
Gray Pagedown	Shift F16 (Next Screen)	^[[6~
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Numlock	PF1	^[OP
Pad /	PF2	^[OQ
Pad *	PF3	^[OR
Pad -	PF4	^[OS
Pad +	+	+
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4

Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+F1	F11	^[[23~
Shift+F2	F12	^[[24~
Shift+F3	F13	^[[25~
Shift+F4	F14	^[[26~
Shift+F5	F15	^[[28~
Shift+F6	F16	^[[29~
Shift+Enter	Enter	^M
Shift+Gray Delete	Character Delete	Δ
Shift+Gray Home	Home Cursor	^[[H
Shift+Pad +	Keypad MINUS	-
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Character Delete	Δ
Shift+Pad End	Line Clear	^[[K
Shift+Pad Down	Cursor Down	^[[B
Shift+Pad Pagedown	Clear Screen	^[[H^[[2J
Shift+Pad Left	Cursor Left	^[[D
Shift+Pad Right	Cursor Right	^[[C
Shift+Pad Home	Home Cursor	^[[H
Shift+Pad Up	Cursor Up	^[[A
Shift+Pad Pageup	Delete Line	^[[M
Control+F1	Shift F1	^[[4i
Control+F2	Shift F2	^[[?1i
Control+F3	Shift F3	^[[0i
Control+F4	Insert Line	^[[L
Control+F5	Line Clear	^[[K
Control+F6	Shift F6	^[[31~
Control+F7	Shift F7	^[[32~
Control+F8	Shift F8	^[[33~
Control+F9	Shift F9	^[[34~
Control+F10	Shift F10	^[[35~
Control+Backspace	Character Delete	Δ
Control+Enter	Enter	^M
Control+Gray Home	Insert Line	^[[L
Control+Gray End	Erase End of Line	^[[K
Control+Gray Pageup	Delete Line	^[[M
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Gray Up	Application Cursor Up	^[OA
Control+Gray Down	Application Cursor Down	^[OB
Control+Gray Left	Application Cursor Left	^[OD
Control+Gray Right	Application Cursor Right	^[OC
Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow

Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application .	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot
Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 9	^[Oy
Shift+Control+F1	Find	^[[1~
Shift+Control+F2	Select	^[[2~
Shift+Control+F3	Insert Here	^[[3~
Shift+Control+F4	Remove	^[[4~
Shift+Control+F5	Prev Screen	^[[5~
Shift+Control+F6	Next Screen	^[[6~
Shift+Control+Pad Down	Application Cursor Down	^[OB
Shift+Control+Pad Left	Application Cursor Left	^[OD
Shift+Control+Pad Right	Application Cursor Right	^[OC
Shift+Control+Pad Home	Insert Line	^[[L
Shift+Control+Pad Up	Application Cursor Up4	^[OA

Wyse 100

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	Function 1 (F1)	^A@^M
F2	Function 2 (F2)	^AA^M
F3	Function 3 (F3)	^AB^M
F4	Function 4 (F4)	^AC^M
F5	Function 5 (F5)	^AD^M
F6	Function 6 (F6)	^AE^M
F7	Function 7 (F7)	^AF^M
F8	Function 8 (F8)	^AG^M
F9	Line Insert	^[E
F10	Line Delete	^[R
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Insert Off	^[r
Gray Delete	Insert Space Character	^[Q
Gray Home	Home Cursor	^^
Gray End	Erase End of Line	^[T
Gray Pageup	Scrl Up	^[w
Gray Pagedown	Scrl Down	^[v
Gray Up	Cursor Up	^K
Gray Down	Cursor Down	^J
Gray Left	Cursor Left	^H
Gray Right	Cursor Right	^L
Pad /	/	/
Pad *	*	*
Pad -	-	-
Pad +	+	+

Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8
Pad 9	9	9
Pad Delete	Insert Space Character	^[Q
Pad Insert	Insert Off	^[r
Pad End	Erase End of Line	^[T
Pad Down	Cursor Down	^J
Pad Pagedown	Scrl Down	^[v
Pad Left	Backspace	^H
Pad Right	Cursor Right	^L
Pad Home	Home Cursor	^^
Pad Up	Cursor Up	^K
Pad Pageup	Scrl Up	^[w
Shift+Tab	Back tab	^[I
Shift+F1	Function 9 (Shift F1)	^AH^M
Shift+F2	Function 10 (Shift F2)	^AI^M
Shift+F3	Function 11 (Shift F3)	^AJ^M
Shift+F4	Function 12 (Shift F4)	^AK^M
Shift+F5	Function 13 (Shift F5)	^AL^M
Shift+F6	Function 14 (Shift F6)	^AM^M
Shift+F7	Function 15 (Shift F7)	^AN^M
Shift+F8	Function 16 (Shift F8)	^AO^M
Shift+F9	Shift Line Insert	^[t
Shift+Gray Delete	Delete Character	^[W
Shift+Gray Home	Cursor to Start of Text	^[{
Shift+Gray End	Page Erase	^[Y
Shift+Gray Pageup	Insert On	^[q
Shift+Pad Delete	Delete Character	^[W
Shift+Pad Insert	Insert On	^[q
Shift+Pad Left	Erase End of Page	^[Y
Shift+Pad Home	Cursor to Start of Text	^[{
Control+Backspace	Rubout	Δ
Control+Enter	Enter	^M
Control+Pad /	/	/
Control+Pad *	*	*
Control+Pad -	-	-
Control+Pad +	+	+
Control+Pad Enter	Enter	^M
Control+Pad .	.	.
Control+Pad 0	Insert Space Character	^[Q
Control+Pad 1	Line Erase	^[T
Control+Pad 2	Cursor Down	^J
Control+Pad 3	Scrl Down	^[v
Control+Pad 4	Backspace	^H
Control+Pad 6	Cursor Right	^L
Control+Pad 7	Home Cursor	^^
Control+Pad 8	Cursor Up	^K

Control+Pad 9	Scrl Up	^[w
Control+Pad Delete	.	.
Control+Pad Insert	0	0
Control+Pad End	1	1
Control+Pad Down	2	2
Control+Pad Pagedown	3	3
Control+Pad Left	4	4
Control+Pad Clear	5	5
Control+Pad Right	6	6
Control+Pad Home	7	7
Control+Pad Up	8	8
Control+Pad Pageup	9	9

Wyse 185

QmodemPro Key	Terminal Key	Characters Sent
Tab	Tab	^I
F1	PF1	^[OP
F2	PF2	^[OQ
F3	PF3	^[OR
F4	PF4	^[OS
F5	Next Screen	^[[6~
F6	F6	^[[17~
F7	F7	^[[18~
F8	F8	^[[19~
F9	F9	^[[20~
F10	F10	^[[21~
Backspace	Backspace	^H
Enter	Enter	^M
Gray Insert	Insert Here	^[[2~
Gray Delete	Remove	^[[3~
Gray Home	Find	^[[1~
Gray End	Select	^[[4~
Gray Pageup	Prev Screen	^[[5~
Gray Pagedown	Next Screen	^[[6~
Gray Up	Cursor Up	^[[A
Gray Down	Cursor Down	^[[B
Gray Left	Cursor Left	^[[D
Gray Right	Cursor Right	^[[C
Numlock	PF1	^[OP
Pad /	PF2	^[OQ
Pad *	PF3	^[OR
Pad -	PF4	^[OS
Pad +	Keypad MINUS	-
Pad Enter	Enter	^M
Pad .	.	.
Pad 0	0	0
Pad 1	1	1
Pad 2	2	2
Pad 3	3	3
Pad 4	4	4
Pad 5	5	5
Pad 6	6	6
Pad 7	7	7
Pad 8	8	8

Pad 9	9	9
Pad Delete	.	.
Pad Insert	0	0
Pad End	1	1
Pad Down	2	2
Pad Pagedown	3	3
Pad Left	4	4
Pad Clear	5	5
Pad Right	6	6
Pad Home	7	7
Pad Up	8	8
Pad Pageup	9	9
Shift+F1	F11	^[[23~
Shift+F2	F12	^[[24~
Shift+F3	F13	^[[25~
Shift+F4	F14	^[[26~
Shift+F5	F15	^[[28~
Shift+F6	F16	^[[29~
Shift+F7	F17	^[[31~
Shift+F8	F18	^[[32~
Shift+F9	F19	^[[33~
Shift+F10	F20	^[[34~
Shift+Enter	Enter	^M
Shift+Gray Delete	Delete	Δ
Shift+Gray Home	Home Cursor	^[[H
Shift+Pad /	/	/
Shift+Pad *	*	*
Shift+Pad -	-	-
Shift+Pad +	+	+
Shift+Pad Enter	Enter	^M
Shift+Pad Delete	Delete	Δ
Shift+Pad End	Erase End of Line	^[[K
Shift+Pad Down	Cursor Down	^[[B
Shift+Pad Pagedown	Clear Screen	^[[H^[[2J
Shift+Pad Left	Cursor Left	^[[D
Shift+Pad Right	Cursor Right	^[[C
Shift+Pad Home	Home Cursor	^[[H
Shift+Pad Up	Cursor Up	^[[A
Shift+Pad Pageup	Delete Line	^[[M
Control+F1	Find	^[[1~
Control+F2	Insert Here	^[[2~
Control+F3	Remove	^[[3~
Control+F4	Select	^[[4~
Control+F5	Prev Screen	^[[5~
Control+Backspace	Delete	Δ
Control+Enter	Enter	^M
Control+Gray Insert	Insert Line	^[[L
Control+Gray Delete	Delete Line	^[[M
Control+Gray End	Erase End of Line	^[[K
Control+Gray Pagedown	Clear Screen	^[[H^[[2J
Control+Gray Up	Application Cursor Up	^[OA
Control+Gray Down	Application Cursor Down	^[OB
Control+Gray Left	Appplication Cursor Left	^[OD
Control+Gray Right	Application Cursor Right	^[OC
Control+Pad /	/	/
Control+Pad *	*	*

Control+Pad -	Keypad Application MINUS	^[Om
Control+Pad +	Keypad Application COMMA	^[Ol
Control+Pad Enter	Keypad Application Enter	^[OM
Control+Pad .	Keypad Application .	^[On
Control+Pad 0	Keypad Application 0	^[Op
Control+Pad 1	Keypad Application 1	^[Oq
Control+Pad 2	Keypad Application 2	^[Or
Control+Pad 3	Keypad Application 3	^[Os
Control+Pad 4	Keypad Application 4	^[Ot
Control+Pad 5	Keypad Application 5	^[Ou
Control+Pad 6	Keypad Application 6	^[Ov
Control+Pad 7	Keypad Application 7	^[Ow
Control+Pad 8	Keypad Application 8	^[Ox
Control+Pad 9	Keypad Application 9	^[Oy
Control+Pad Delete	Keypad Application .	^[On
Control+Pad Insert	Keypad Application 0	^[Op
Control+Pad End	Keypad Application 1	^[Oq
Control+Pad Down	Keypad Application 2	^[Or
Control+Pad Pagedown	Keypad Application 3	^[Os
Control+Pad Left	Keypad Application 4	^[Ot
Control+Pad Clear	Keypad Application 5	^[Ou
Control+Pad Right	Keypad Application 6	^[Ov
Control+Pad Home	Keypad Application 7	^[Ow
Control+Pad Up	Keypad Application 8	^[Ox
Control+Pad Pageup	Keypad Application 8	^[Oy
Shift+Control+Pad Down	Application Cursor Down	^[OB
Shift+Control+Pad Left	Application Cursor Left	^[OD
Shift+Control+Pad Right	Application Cursor Right	^[OC
Shift+Control+Pad Home	Insert Line	^[[L
Shift+Control+Pad Up	Application Cursor Up	^[OA

Command Overview

This is a complete list of menu commands that appear in *QmodemPro for Windows.* To see help about a specific menu command while youre running *QmodemPro for Windows*, hold down the left mouse button on the menu command youre interested in, and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

QmodemPro Window

Fax Viewer Window

File Viewer Window

GIF Viewer Window

Phonebook Window

Script Debugger Window

Text Editor Window

QmodemPro Window

This is a complete list of menu commands that appear in *QmodemPro for Windows*. To see help about a specific menu command while youre running *QmodemPro for Windows*, hold down the left mouse button on the menu command youre interested in, and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

<u>File</u>

<u>Upload</u>

<u>Download</u>

<u>View a File</u>

<u>View GIF File</u>

<u>Edit a File</u>

<u>Capture</u>

<u>Log Toggle</u>

<u>Log View</u>

<u>Printer Toggle</u>

<u>Shell to DOS</u>

<u>Exit</u>

<u>Edit</u>

<u>Copy</u>

<u>Paste</u>

<u>Screen to</u>   <u>Clipboard</u>

             <u>Capture File</u>

             <u>File</u>

             <u>Printer</u>

<u>Scrollback to</u>   <u>Clipboard</u>

             <u>Capture File</u>

             <u>File</u>

             <u> Printer</u>

<u>Phonebook</u>

<u>Directory</u>

<u>Manual Dial</u>

<u>Fax</u>

<u>Receive Log</u>

<u>Auto Receive</u>

<u>Receive</u>

<u>Send</u>

Device

Select

Fine Tune

Hangup

Break

XonXoff

Full Duplex

Reset Modem

Auto Answer On

Auto Answer Off

Terminal

Emulation

Clear Screen

Reset Emulation

Doorway Mode

Add LF to CR

8th Bit Strip

Split Screen

Scrollback

Scrollback Record

Scripts

Execute

Quicklearn

Edit

Compile

Debug

Host Mode Script

Options

Files: Path Definitions

            File Definitions

            Executable Programs

Phonebook

Device

Fax

Emulations

Protocols

Sounds

Desktop

Macro Keys

Help

Tool Bar

Dialer

Upload

Download

Review

Capture

Printer

Device

Fine Tune

Hangup

Run Script

Fax Viewer Window

This is a complete list of menu commands that appear in the *QmodemPro for Windows*. **Fax Viewer**. To see help about a specific menu command while youre running *QmodemPro for Windows*, hold down the left mouse button on the menu command youre interested in, and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

<u>File</u>

<u>Print</u>

<u>Print Setup</u>

<u>Exit</u>

<u>Edit</u>

<u>Copy</u>

<u>Page</u>

<u>Zoom</u>

<u>Next Page</u>

<u>Previous Page</u>

<u>View Pages</u>

<u>Help</u>

<u>Tool Bar</u>

<u>Print</u>

<u>Copy</u>

<u>25%</u>

<u>50%</u>

<u>100%</u>

<u>Previous</u>

<u>Next</u>

<u>Pages</u>

File Viewer Window

This is a complete list of menu commands that appear in the *QmodemPro for Windows* **File Viewer**. To see help about a specific menu command while youre running *QmodemPro for Windows*, hold down the left mouse button on the menu command youre interested in, and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

File

Open

Print

Print Setup

Exit

Edit

Copy

Select All

Search

Find

Find Again

Goto Line

Options

Font

Hex Mode

Tab Expansion

Tab Size

Help

Tool Bar

Open

Print

Copy

Find

Find Next

GIF Viewer Window

This is a complete list of menu commands that appear in the *QmodemPro for Windows* **GIF Viewer**. To see help about a specific menu command while youre running *QmodemPro for Windows*, hold down the left mouse button on the menu command youre interested in, and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

<u>File</u>

<u>Open</u>

<u>Delete</u>

<u>Print</u>

<u>Print Setup</u>

<u>Exit</u>

<u>Edit</u>

<u>Copy</u>

<u>View</u>

<u>Zoom</u>

<u>Full Screen</u>

<u>Next</u>

<u>Previous</u>

<u>Stop</u>

<u>Help</u>

<u>Tool Bar</u>

<u>Open</u>

<u>Print</u>

<u>Copy</u>

<u>Zoom</u>

<u>Screen</u>

<u>Next</u>

<u>Previous</u>

<u>Stop</u>

Phonebook Window

This is a complete list of menu commands that appear in the *QmodemPro for Windows* **Phonebook**. To see help about a specific menu command while youre running *QmodemPro for Windows*, hold down the left mouse button on the menu command youre interested in, and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

File

Open

Pack

Print

Print Setup

Exit

Edit

Cut

Copy

Paste

Append

Add Entry

Insert Entry

Delete Entry

Revise

Group

Sort

Search

Find

Find Next

Goto Entry

Dial

Dial

Manual Dial

View

List

Icon

Icon List

Help

Tool Bar

[Dial](#)

[Cut](#)

[Copy](#)

[Paste](#)

[Add](#)

[Delete](#)

[Revise](#)

[Find](#)

[Find Next](#)

[Group](#)

[Print](#)

[Sort](#)

Script Debugger Window

This is a complete list of menu commands that appear in the *QmodemPro for Windows* **Script Debugger**. To see help about a specific menu command while youre running *QmodemPro for Windows*, hold down the left mouse button on the menu command youre interested in, and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

File

ExitExit_menu_command

Debug

Step

Jump

Return

Go

Stop

Watch

Window

Tile

Cascade

Arrange Icons

Source

Watch

Help

Tool Bar

Step

ump

Return

Go

Stop

Watch

Text Editor Window

This is a complete list of menu commands that appear in the *QmodemPro for Windows* **Text Editor**. To see help about a specific menu command while youre running *QmodemPro for Windows*, hold down the left mouse button on the menu command youre interested in, and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

File

New

Open

Save

Save As

Print

Print Setup

Exit

Edit

Undo

Redo

Cut

Copy

Paste

Select All

Search

Find

Replace

Find/Replace Again

Goto Line

Options

Auto-Indentation

Font

Tab Size

Wordwrap

Wrap at Column

Help

Tool Bar

Open

Save

Save As

Print

Cut

Copy

Paste

Undo

Redo

Find

Replace

Repeat

Debug Menu (Script Debugger)

See Also: <u>SLIQ Script Language Interface for QmodemPro</u>

The **Debug** menu allows you to select various debug operations, for instance Step, Go, Return, Jump, and Watch. For more information on how to write and debug scripts, see Book 2 of the documentation.

Device Menu (QmodemPro Window)

See Also: <u>Options</u>

The **Device** menu allows you to select from any available devices, and **Fine Tune** or change certain settings for the current device. Devices are installed and configured from the **<u>Options/Device</u>** menu.

Dial Menu (Phonebook Window)

The Dial menu offers commands to dial phonebook entries, either individually or in groups, or manually dial a phone number.

Edit Menu

Where: **QmodemPro/Main Menu**
File Viewer/Main Menu
GIF Viewer/Main Menu
Text Editor/Main Menu
Phonebook/Main Menu
Fax Viewer/Main Menu

See Also: **Cut**
Copy
Paste
Add
Insert
Delete
Append

The Edit menu contains commands for copying, cutting and pasting text within a window. Different windows offer different options, appropriate to the kind of operations available in that window. The File Viewer, for instance, allows you **Copy** text to the clipboard, but since you're only viewing and not editing the file in the window, the **Cut** and **Paste** commands are not available. Meanwhile, the Text Editor has Cut and Paste commands, and the Phonebook offers additional Editing selections to add and remove Phonebook entries.

Fax Menu (QmodemPro Window)

See Also: **Fax Options**
Fax Receive Log
Fax Send
Fax Viewer Window

From this menu, you can **Send, Receive**, or **Auto Receive** Fax messages, and view the Fax **Receive Log**.

File Menu

The File Menu contains commands related to file management. Depending on the window, you'll see commands for opening, closing, editing and viewing files, and for closing and exiting the current window.

For instance, the *QmodemPro* file menu has commands to Upload and Download files, View and Edit files, open and close the Capture and Log files, open a DOS Shell window, and more.

Options Menu

Where: **QmodemPro/Main Menu**
            File Viewer/Main Menu
            Text Editor/Main Menu

The main *QmodemPro for Windows* **Options** controls most of the configuration for the program. In other areas, the **Options** menu lists commands which change the way the current window looks and behaves. For instance, in the Text Editor, the Options menu allows you to select a font to display text in the window, turn **on** Wordwrap mode, and set the number of spaces to allocate to a tab character.

Page Menu (Fax Viewer Window)

See Also: **<u>Fax Viewer Window</u>**
<u>Next Page</u>
<u>Previous Page</u>
<u>Zoom</u>

This menu lists commands which page forward or backward in a multi-page Fax document, and enlarge or reduce the image in the window.

Phonebook Menu (QmodemPro Window)

See Also: **Revise Phonebook Entry**
Dial
Manual Dial
Send Fax
The Phonebook

The commands on this menu allow you to open the Phonebook Window, or manually dial a phone number.

Scripts Menu (QmodemPro Window)

See Also: **Chapter 5, The QmodemPro for Windows Host**

This menu offers commands to execute, QuickLearn, compile and debug scripts. You can also start the Host script from here. For more information on scripts, please refer to Book 2 of the documentation.

Search Menu

Where: **Phonebook/Main Menu**
             File Viewer/Main Menu
             Text Editor/Main Menu

See Also: The Editor

The **Search** command allows you to find and highlight text in a file or window exactly matching the search pattern you enter in a text dialog box.

Terminal Menu (QmodemPro Window)

See Also: **<u>Emulation Options</u>**
<u>Customizing the Desktop</u>
<u>Changing the Appearance of the Terminal Screen</u>

This group of commands governs how your local terminal behaves. From here you can select a terminal emulation, clear the screen, view the scrollback buffer, and more.

Tool Bar

Where: **Fax Viewer Window**
File Viewer Window
GIF Viewer Window
Phonebook Window
QmodemPro Window
Script Debugger Window
Text Editor Window

See Also: **Desktop Options**
Changing the appearance of the terminal screen

The Tool Bar is shown in the top portion of each window below the menu bar, and contains buttons for frequently-used menu commands. For instance, the Phonebook Window Tool Bar contains buttons to Dial, Add, Revise entries, Sort, and Print the Phonebook.

View Menu (GIF Viewer Window)

This menu changes the size of the image on your screen.

View Menu (Phonebook Window)

Icon

        Icon List

        List

This menu allows you to select the viewing mode for your Phonebook lists: Icon, Icon List or List.

Window Menu, Script Debugger Window

This menu allows you to rearrange the way parts of the Script Debugger Window are displayed.

8th Bit Strip [menu command]

Where: **QmodemPro/Terminal**

See Also: **Emulation Options**

A check mark appears beside this menu selection when **8th Bit Strip** is toggled on.

Some computer systems, for example CompuServe and GEnie, use the high bit of each 8-bit character for parity checking. If the **high bit** is not removed, the screen display will consist of high-order ASCII characters (ASCII 128+) such as foreign, graphic, or scientific characters.

If you see garbage or foreign characters on your screen, try turning **on** the **Terminal/8th Bit Strip** command.

About QmodemPro ... [menu command]

Where: **QmodemPro/Help**

This menu selection displays an information window about *QmodemPro for Windows*. Your registration information is shown in the lower part of the window.

Add Entry ... [menu command, button]

Where: **Phonebook/Edit Menu** + **Tool Bar**

See Also: **Revise Phonebook Entry**
The Phonebook

Adds an entry to the end of the Phonebook. The **Add Entry** command pops up the **Add Phonebook Entry** dialog box into which you can enter information about the new entry.

Add LF to CR [menu command]

Where: **QmodemPro/Terminal**

See Also: **Emulation Options**
             ASCII Options

A check mark appears beside this menu selection when **Add LF to CR** is toggled **on**.

Various computers differ in how they indicate the end of a line of text. The IBM PC and compatible computers require two characters: a **Carriage Return** plus a **Line Feed** to end a line and move the cursor down to the beginning of the next line.

Meanwhile, other computers use only one character to indicate a new line. The result, if this setting is incorrect, is that text lines will be overwritten on your screen. if this is the case, turn **on** the **Add LF to CR** command from the **Emulation** menu.

Append [menu command]

Where: **Phonebook/Edit**

See Also: **Cut**
            Copy
            Paste

The **Append** command pastes phonebook entries from the Clipboard to the **End** of the Phonebook, rather than at the cursor location.

Arrange Icons

Moves icons back to their proper positions on the screen.

Auto Answer On/Off [menu command]

Where: **QmodemPro/Device**

These commands send the **Answer on** and **Answer off** strings respectively, specified in the Device Configuration for the currently active device. The default for Answer On is ATS0=1^M, while the default for Answer Off is ATS0=0^M.

Auto Receive ... [menu command]

Where: __QmodemPro/Fax__

See Also: __Receive Log__

This menu command turns on **Auto Receive Fax** mode. If your modem is capable of receiving CLASS1 or CLASS2 Fax, *QmodemPro for Windows* will automatically answer Fax calls and receive the incoming Fax messages as files on disk, providing the selected device is configured for Fax..

When Auto Receive is selected the __Receive Log__ dialog box is displayed with the Auto-receive option checked. From that dialog box you can view, print or delete Fax messages.

Auto-Indentation [menu command]

Where: <u>**Text Editor/Options**</u>

Default: OFF

Auto-indentation or "sticky indent" causes the text in a paragraph to line up with the left margin established by the first line in the paragraph. A check mark on the menu indicates that Auto-Indentation is turned **on**.

With auto-indentation **on**, the editor produces an effect something like this:

```
        First line of text indented from left
        margin, followed by more text, and
        more text, and still more text.
```

With auto-indentation turned **off**, the effect looks more like this:

```
         First line of text indented from left margin, followed by more
    text, and more text, and still more text.
```

Break [menu command]

Where: **QmodemPro/Device**

See Also: **Device Connection**
              Fine Tuning

This selection sends a BREAK signal. **Break** is a special signal generated by the serial adapter, used to notify the receiving computer of a specific condition. In most cases, Break is used to tell the other computer to interrupt whatever it is doing.

Capture [menu command, button]

Where: **QmodemPro/File Menu** + **Tool Bar**

See Also: **"Generic" dialog boxes**
Scrollback to
Log Toggle
File Definitions

The Capture File is a recording of an on-line session. As the session progresses, incoming data is written to a text file as well as to the terminal window. When you open the Capture file, a dialog box will pop up prompting you to accept or modify the default Capture file name and path.

Cascade windows

Overlaps the Source and Watch windows

Clear Screen [menu command]

Where: **QmodemPro/Terminal**

This command clears the terminal window, resets the screen colors to the default and returns the cursor to the top left corner of the window.

Command Overview [menu command]

The Windows Help system is extremely comprehensive, and offers an easy way to search for topics that interest you, jump from one topic to another, annotate entries and print help topics. The *QmodemPro for Windows* help system takes full advantage of the Windows help engine.

Command Overview

The Command Overview lists each menu command in order by the menu on which it appears. Click on the text of a topic that interests you for more information.

Compile Script [menu command]

Where: **QmodemPro/Scripts**

See Also: **SLIQ Script Language Interface for QmodemPro**

SLIQ scripts must be compiled to an executable format before they can be used. Compiled scripts are advantageous because they execute faster than interpreted scripts and allow the author to keep the script source code confidential. If an attempt is made to execute a script before it has been compiled, the compiler is automatically invoked by the program. Once compiled, either manually or automatically, a script is executed immediately when requested.

Copy [menu command, button]

Where: **QmodemPro/Edit**
               File Viewer/Edit Menu + Tool Bar
               GIF Viewer/Edit Menu + Tool Bar
               Text Editor/Edit Menu + Tool Bar
               Phonebook/Edit Menu + Tool Bar
               Fax Viewer/Edit Menu + Tool Bar

See Also: **Cut**
               Paste
               Viewing a File
               The Editor

Copies the current selection to the Windows clipboard. Note that anything you copy to the clipboard will *replace* the current clipboard contents.

Cut [menu command, button]

Where: <u>**Text Editor/Edit Menu**</u> **+** <u>**Tool Bar**</u>
                   <u>**Phonebook/Edit Menu**</u> **+** <u>**Tool Bar**</u>

See Also: <u>**Copy**</u>
                    <u>**Paste**</u>

Copies the selected text to the clipboard, and deletes the text from the document. Note that anything you copy to the clipboard will *replace* the current clipboard contents.

Debug Script [menu command]

Where: **QmodemPro/Scripts**

See Also: **SLIQ Script Language Interface for QmodemPro**
Script Debugger Window

Starting the SLIQ debugger is done by selecting the **Script/Debug** menu choice, which displays the standard file selection dialog box. Once a script file is selected, it is recompiled, and the **Script Debugger Window** pops up, allowing you to analyze the logic and operation of your script.

Delete [menu command]

Where: **GIF Viewer/File**

Use this command to **delete** the file you are currently viewing, from the disk. If you downloaded a GIF picture and didn't like it, this is an easy way to get rid of it and free up the disk space it would have occupied.

Delete Entry [menu command, button]

Where: **Phonebook/File Menu** + **Tool Bar**

Deletes (removes) the currently selected Phonebook entry or entries. A confirmation window will pop up before the entries are removed. Note that this deletion is permanent!

Desktop ... [menu command]

Where: **QmodemPro/Options**

See Also: **Desktop Options**
                Emulation Options
                Macro Keys
                Customizing the desktop
                Changing the appearance of the terminal screen

The Desktop menu allows you to assign colors, patterns and other options to the desktop surrounding the terminal window.

Device ... [menu command]

Where: **QmodemPro/Options**

See Also: **Edit Devices**
Devices
Other Communication Devices

Opens the **Edit Devices** dialog box. From this dialog box, you can select, add and modify devices associated with one or more modems connected to your PC.

Device [button]

Where: **QmodemPro/Tool Ba**

See Also: **Select Device**
              Devices
              Other Communication Devices

Opens the **Select Device** dialog box. This button allows you to select a device (modem) to use for the current operation.

Dial [menu command, button]

Where: **Phonebook/Dial Menu** + **Tool Bar**

See Also: **Dialer**
The Phonebook

Immediately starts to dial any entry or entries tagged in the Phonebook. The **Dialer** dialog box will pop up, showing the list of systems to dial, and the progress of the current call. When a connection is established, the Dialer dialog box disappears and *QmodemPro for Windows* goes into Terminal mode.

Dialer [button]

Where: **QmodemPro/Tool Bar**

See Also **Phonebook**

Opens the **Phonebook** window.

Directory ... [menu command]

Where: __QmodemPro/Phonebook__

See Also __Phonebook__

Opens the **__Phonebook__** window.

Doorway Mode [menu command]

Where: <u>**QmodemPro/Terminal**</u>

See Also: <u>**Emulation Options**</u>
            <u>**Emulation Keyboard Mapping**</u>

Default: OFF

Doorway mode allows you to send the entire PC Keyboard Scan Codes across the line to the remote system. This is most often used with BBS systems that make full use of the arrow keys and **[PgUp]** and **[PgDn]**. You can go into Doorway mode from the **Terminal/Doorway mode** menu command. Note that doorway mode can only be used if the system you are calling also has installed support for the doorway emulation.

A check mark appears beside this menu selection when Doorway Mode is toggled on. The remote host can often turn Doorway Mode on in *QmodemPro* automatically when needed.

Download ... [menu command, button]

Where: **QmodemPro/File Menu** + **Tool Bar**

See Also: **Select Download Protocol**
Path Definitions
Protocol Options
Uploading and Downloading Files
What to do with Compressed Files
Viewing a File

Downloading refers to copying one or more files from a host computer such as a BBS, or an On-line Service, onto your computer. Normally, the **Download** command first opens the **Select Download Protocol** dialog box, and then prompts for a download path. Some file transfer protocols, such as Zmodem and CIS B+, can start automatically, bypassing the path selection. Other protocols, such as the various flavors of Xmodem, require you to enter a file name as well as a path before beginning the file transfer.

The default download path is taken from the **Options/Path Definitions/Download Path** dialog box entry.

Edit a File ... [menu command]

Where: **QmodemPro/File**

See Also: **The Editor**

QmodemPro for Windows contains a text editor suitable for creating and editing scripts, text files, batch files, and other files which need to be edited and saved as plain ASCII text.

You will be prompted to select a file to edit, which will then be loaded into the Text Editor window.

Edit Script [menu command]

Where: **QmodemPro/Scripts**

See Also: **Edit a File ... [menu command]**
           SLIQ Script Language Interface for QmodemPro
           Path Definitions [dialog box]
           QuickLearn ... [menu command]

This menu command loads the script file you select into the text editor. The file selection dialog box lists files with the extention .SCR, in your script files directory.

Emulation ... [menu command]

Where: **QmodemPro/Terminal**

See Also: **Emulation Options**

Terminal Emulation allows *QmodemPro for Windows* to behave like (emulate) a terminal other than the standard IBM compatible PC. Over 30 different terminal emulations are available in *QmodemPro for Windows*. When you select a terminal emulation, keyboard and video information is interpreted so that your PC behaves as if it were a native terminal of the selected type.

Emulations ... [menu command]

Where: **QmodemPro/Options**

See Also: **Phonebook**
Emulation Font
Emulation Translation Table
Emulation Color
Emulation Keyboard Mapping
RIPscrip Options
Changing the appearance of the terminal screen
Emulations
Programmable Keys

Opens the **Emulation Options** dialog box, allowing you to edit terminal emulation settings such as Emulation Keyboard Mapping, color, translation, and more.

Execute Script [menu command]

Where: **QmodemPro/Scripts**

See Also: **Edit a File ... [menu command]**
                SLIQ Script Language Interface for QmodemPro
                Path Definitions [dialog box]
                QuickLearn ... [menu command]

Loads, compiles (if necessary), then executes the selected script.

Exit [menu command]

Where: **QmodemPro/File**
File Viewer/File
GIF Viewer/File
Text Editor/File
Phonebook/File
Fax Viewer/File

Closes the current window or application. In the **QmodemPro** window, a warning box will pop up if you are on line when you select the **Exit** command.

Fax ... [menu command]

Where: **QmodemPro/Options**

See Also: **Fax Options**
Fax Receive Log
Fax Send
Fax Viewer Window

This command pops up the **Fax Options** dialog box. The Fax options determine the appearance, arrangement and content of Faxes sent by *QmodemPro for Windows.* Normally, a Fax consists of a cover sheet identifying the sender and addressee, plus the actual text or graphic file being sent.

Files: Executable Programs ... [menu command]

Where: **QmodemPro/Options**

See Also: **Executable Programs**
                View a File
                Edit a file

This menu pops up the **Executable Programs** dialog box and allows you to substitute your own file viewer and editor programs in place of the ones provided with *QmodemPro for Windows.*

Files: File Definitions ... [menu command]

Where: **QmodemPro/Options**

See Also: **File Definitions**
Log View
Capture

This menu pops up the File Definitions dialog box and allows you to assign default file names for files automatically recorded by *QmodemPro for Windows.*

Files: Path Definitions ... [menu command]

Where: **QmodemPro/Options**

See Also: Path Definitions
             Upload
             Download

This menu pops up the **Path Definitions** dialog box and lets you assign default paths for files used by *QmodemPro for Windows.*

Find ... [menu command, button]

Where: **File Viewer/Search Menu** + **Tool Bar**
                      Text Editor/Search Menu + **Tool Bar**
                      Phonebook/Search Menu + **Tool Bar**

See Also: **Find Phonebook Entry**
                      Find Text

The command pops up a **Find** dialog box and allows you to define the text to search for, and search options.

Find Next [menu command, button]

Continues the **search** operation, using the data already in the "Text to find" field in the **Find** dialog box.

Fine Tune ... [menu command, button]

Where: **QmodemPro/Device Menu + Tool Bar**

See Also: **Device Settings**
Options
Devices
Fine Tuning

The **Fine Tune** menu pops up the **Device Settings** dialog box allowing you to make temporary changes to the current device configuration. This dialog box has options to change baud rate, parity, stop bits, and data bits.

Font ... [menu command]

Where: **File Viewer/Options**
              Text Editor/Options

See Also: **View a File**
              Edit a File
              Viewing a File
              The Text Editor

Pops up the **Select Font** dialog box, which allows you to change the type face, style and size of the text in the File Viewer or Text Editor window.

Full Duplex [menu command]

Where: **QmodemPro/Device**

See Also: Device Connection

Duplex describes the method used to display your keystrokes on the terminal screen. In Full Duplex mode, the host system echoes your keystrokes back to you. In Half Duplex mode, the host system does not echo your keystrokes, so *QmodemPro for Windows* must do it instead.

A check mark appears beside this menu selection when Full Duplex is toggled **on**.

Full Screen [menu command]

Where: **GIF Viewer/View**

See Also: Screen

Expands the GIF viewer window to full screen. Click anywhere on the screen to restore the window to its previous dimensions.

Go [menu command, button]

Where: **Script Debugger/Debug**

See Also: **SLIQ Script Language Interface for QmodemPro**
Return
Step
Stop
Jump

The Go button continuously executes the script until the script ends or a breakpoint is encountered.

Goto Entry ... [menu command]

Where: **Phonebook/Search**

Pops up a the **Goto Entry** dialog box, allowing you to go directly to a phonebook entry if you know the entry number in the phonebook file.

Goto Line ... [menu command]

Where: **File Viewer/Search**
Text Editor/Search

See Also: **Search**

Pops up the **Goto Line** dialog box, allowing you to go directly to the specified line number in the file you're viewing or editing.

Group ... [menu command, button]

Where: **Phonebook/File Menu** + **Tool Bar**

See Also: **Edit Group**
            Edit Groups

Pops up the **Edit Groups** dialog box, allowing you to tag and dial multiple Phonebook entries in a single pass.

Hangup [menu command, button]

Where: **QmodemPro/Device Menu** + **Tool Bar**

See Also: **Device Modem**

The Hang up command forces the modem to terminate a call. In most cases, you would *not* use this command to disconnect from the host computer; you would use the host computer's standard logoff procedure. However, in case you are experiencing problems such as remote computer lockup, or bad line noise, you may use this command to disconnect the call.

Help Contents [menu command]

The Windows Help system is extremely comprehensive, and offers an easy way to search for topics that interest you, jump from one topic to another, annotate entries and print help topics. The *QmodemPro for Windows* help system takes full advantage of the Windows help engine.

Contents

The Contents lists all the available help topics, including menu commands, dialog boxes, How-To information and a troubleshooting guide. Click on the text of a topic that interests you for more information.

Help Index [menu command]

The Windows Help system is extremely comprehensive, and offers an easy way to search for topics that interest you, jump from one topic to another, annotate entries and print help topics. The *QmodemPro for Windows* help system takes full advantage of the Windows help engine.

Index

The Index lists each menu command in alphabetical order. Click on the text of a topic that interests you for more information.

Hex mode [menu command]

Where: **File Viewer/Options**

See Also: **File Viewer**

Displays the hexadecimal (Hex) values of the characters in the current file.

Host Mode Script [menu command]

Where: __QmodemPro/Scripts__

See Also: Chapter 5, The QmodemPro for Windows Host
__SLIQ Script Language Reference__

Loads and executes HOST.SCR, the *QmodemPro for Windows* Host mode. The Host mode allows you to receive incoming data calls as if it were a scaled-down Bulletin Board System. Callers can send and receive messages, and transfer files. See Chapter 5 for complete details.

How to Use Help [menu command]

The Windows Help system is extremely comprehensive, and offers an easy way to search for topics that interest you, jump from one topic to another, annotate entries and print help topics. The *QmodemPro for Windows* help system takes full advantage of the Windows help engine.

How to Use Help

Pops up the help screen for the Windows help system. This series of screens explains how to use Windows help.

Icon [menu command]

Where: **Phonebook/View**

See Also: **Revise Phonebook Entry**
Icon List
List

Lists Phonebook entries as Icons rather than text or text plus icons. Double clicking on an icon dials that entry.

Icon List [menu command]

Where: **Phonebook/View**

See Also: **Revise Phonebook Entry**
          Icon
          List

Lists Phonebook entries as Icons plus Text rather than Icons only or Text only. Double clicking on an icon or text entry dials that entry.

Insert Entry ... [menu command]

Where: **Phonebook/File**

See Also: **Revise Phonebook Entry**

Inserts a new entry in the phonebook above the currently highlighted entry. The **Revise Phonebook Entry** dialog box pops up.

Jump

Where: **Script Debugger/Debug**

See Also: **SLIQ Script Language Interface for QmodemPro**
Return
Step
Stop

The Jump button is similar to the Step button, except that if the current statement calls a subroutine or function, the statements within the subroutine or function will automatically be skipped.

List [menu command]

Where: **Phonebook/View**

See Also: **Revise Phonebook Entry**
               Icon List
               Icon

Lists Phonebook entries as Text only. Double clicking on an entry dials that entry.

Log Toggle [menu command]

Where: **QmodemPro/File**

See Also: **Capture**
            File Definitions

This command starts or stops the **Session Log**. The log is used to keep track of major *QmodemPro for Windows* activity, such as uploading and downloading, the names and numbers of systems you dialed, date and time of the call, and how long the on-line session lasted.

Log View [menu command]

Where: <u>QmodemPro/File</u>

See Also: <u>Log Toggle</u>
<u>File Viewer</u>
<u>File Definitions</u>

Opens a file selection dialog box with the name of the defined log file as the default entry.
<u>The</u> **File Viewer** displays the selected log file.

Macro Keys ... [menu command]

Where: **QmodemPro/Options**

See Also: **Emulation Keyboard Mapping**
Desktop Options
Programmable Keys

Opens the **Macro Keys** dialog box, allowing you to assign macros and commands to the numeric keys in the **[Alt]**, **[Alt][Shift]**, **[Alt][Ctrl]** and **[Alt][Shift][Ctrl]** positions.

Manual Dial ... [menu command]

Where: **QmodemPro/Phonebook**
            Phonebook/Dial

See Also: **Phonebook**

The **Manual Dial** command pops up the **Manual Dial** dialog box, allowing you to dial a number that is not in your Phonebook. Manual dial commands may contain up to 40 alphabetic or numeric characters.

New [menu command]

Where: <u>**Text Editor/File**</u>

Clears the window and opens a new, untitled file for input and editing. You will be prompted to save any changes to the previous file you were working on before the old file is closed.

Next [button]

Where: **GIF Viewer/Tool Bar**

When multiple files have been selected in the **Open GIF file** dialog box, this button loads the next GIF picture into the Viewer for viewing.

Next Page [menu command]

Where: **Fax Viewer/Page**

See Also: **Previous Page**

Displays the next page of a multi-page Fax message.

Open ... (Phonebook) [menu command, button]

Where: **Phonebook/File**

See Also: **Phonebook**
              Phonebook Options

This menu selection allows you to load a different phonebook file. A common dialog box will pop up allowing you to browse for file names, paths and drives.

Open ... [menu command, button]

Where: <u>File Viewer/File Menu</u> **+** <u>Tool Bar</u>
                 <u>GIF Viewer/File Menu</u> **+** <u>Tool Bar</u>
                 <u>Text Editor/File Menu</u> **+** <u>Tool Bar</u>

Opens a dialog box prompting you for the drive, path and name of the file you wish to **open** for viewing or editing.

Pack [menu command]

Where: **Phonebook/File Menu**

This command removes deleted record space from the Phonebook file. Use this command to reduce the size of the Phonebook file on disk after you delete a large number of entries

Paste [menu command, button]

Where: **QmodemPro/Edit**
Text Editor/Edit Menu + **Tool Bar**
Phonebook/File Menu + **Tool Bar**

See Also: **Cut**
Copy
The Phonebook
The Editor

Copies the contents of the clipboard to the document, at the cursor location.

Phonebook ... [menu command]

Where: **QmodemPro/Options**

See Also: **Phonebook Options**
Phonebook Window
The Phonebook

This menu command pops up the **Phonebook Options** dialog box, allowing you to specify the name and path of the default phonebook, and whether or not to open the Phonebook Window each time you start *QmodemPro for Windows.*

Previous [button]

Where: <u>**GIF Viewer/Tool Bar**</u>

Loads the previous GIF picture file back into the GIF viewer for viewing.

Previous Page [menu command]

Where: **Fax Viewer/Page**

See Also: **Next Page**

Displays the previous page of a multi-page Fax message.

Print [menu command, button]

Where: **File Viewer/File**
GIF Viewer/File
Text Editor/File

See Also: **Print**
Print Phonebook [dialog box]
Print Setup

Pops up the Print Dialog Box, allowing the text or graphics in the current window to be printed.

Print Phonebook [menu command, button]

Where: **Phonebook/File**

See Also: **Print**
Print Phonebook [dialog box]
Print Setup

Pops up the Print Phonebook dialog box, allowing you to select phonebook entries to print.

Print Setup [menu command]

Where: **Phonebook/File**

See Also: **Print**
Print Phonebook [dialog box]
Print Setup

Pops up the Print Setup dialog box, allowing you to select and configure your printer before printing text or graphics.

Printer Toggle [menu command, button]

Where: **QmodemPro/File Menu** + **Tool Bar**

See Also: **Capture**
File Definitions

Default: Off

This is a simple **on**-**off** selection. When the **printer** is toggled **on**, data is sent to the currently selected printer in Windows as well as to the terminal window. When it is toggled **off**, data is sent to the terminal window only.

Protocols ... [menu command]

Where: **QmodemPro/Options**

See Also: **Protocol Options**

Pops up the **Protocol Options** dialog box.

QuickLearn ... [menu command]

Where: __QmodemPro/Scripts__

This menu command turns **on** the *QmodemPro for Windows* script recorder. For more information on scripts, please refer to Book 2 of this documentation.

Receive ... [menu command]

Where: <u>**QmodemPro/Fax**</u>

See Also: <u>**Receive Log**</u>
    <u>**Auto-Receive**</u>
    <u>**Fax Viewer Window**</u>
    <u>**Path Definitions**</u>

Puts *QmodemPro for Windows* into manual Receive Fax mode and pops up the **Fax Receive Status** dialog box. The phone call will be answered immediately by *QmodemPro* in Fax mode, and the incoming Fax will be saved in the **Receive Fax** path specified in the <u>**Path Definitions**</u> dialog box.

This command runs once, then exits back to the terminal window. To receive Faxes automatically, use the **Auto Receive** command.

Receive Log ... [menu command]

Where: **QmodemPro/Fax**
See Also: **Auto Receive**
Fax Viewer Window

QmodemPro for Windows stores received Fax messages as files on disk. This menu command pops up a dialog box listing Faxes received and stored on disk. The list shows the **date** and **time** the Fax was received, the number of **pages**, and the **identifier**, usually the phone number of the originating Fax machine. From this dialog box, you can highlight a file in the list, then **View** the incoming Fax messages on screen, **Print** a Fax, or **Delete** a Fax from your disk.

To receive Faxes automatically, click on the **Auto Receive** check box. When a phone call comes in, *QmodemPro for Windows* will attempt to establish a Fax connection. If it is successful, it will receive the incoming Fax and store it on disk for later review.

Redo [menu command, button]

Where: **Text Editor/Edit Menu** + **Tool Bar**

See Also: **The Editor**

Restores the results of the last editing command removed with the Undo command. It "undoes" an "undo".

Repeat [button]

Where: __Text Editor/Tool Bar__

See Also: __Find Text__
     __Replace Text__
     __The Editor__

Repeats the current **Find** or **Replace** operation using the text entered in the __Find Text__ or __Replace Text__ dialog boxes.

Replace ... [menu command, button]

Where: <u>**Text Editor/Search Menu**</u> **+** <u>**Tool Bar**</u>

See Also: <u>**The Editor**</u>

The **Replace** dialog box allows you to define the text to search for, the text you want to replace it with, and search options.

Reset Emulation [menu command]

Where: **QmodemPro/Terminal**

See Also: **Clear Screen**
            Emulation Options

Clears the screen and homes the cursor, empties the scrollback buffer, resets color attributes to the default for the current emulation. If you're currently having trouble making sense of what you see on your screen, particularly if the current session started out all right, try the Reset Emulation command.

Reset Modem [menu command]

Where: **QmodemPro/Device**

See Also: **Device Options**
Device Modem

Sends the Modem Initialization string to the current device. Use this command to reinitialize your modem if you were disconnected from a host because of line noise, or if your modem isn't responding the way you expect.

Return [menu command, button]

Where: **Script Debugger/Debug**

See Also: **SLIQ Script Language Interface for QmodemPro**
Jump
Step
Stop

The Return button causes the debugger to execute statements continuously until the return from the current subroutine or function. This button is useful if you Stepped into a subroutine or function and meant to Jump over it.

Review [button]

Where: **QmodemPro/Tool Bar**

See Also: **Scrollback**
            Recording Your Session

Goes into **Scrollback** mode. The Scrollback buffer contains a recording of your current session. Use the arrow keys, **[PgUp]** and **[PgDn]** down keys to page through the scrollback, or use the scroll bars on the side of the window.

Revise ... [menu command, button]

Where: **Phonebook/File Menu** + **Tool Bar**

See Also: **Phonebook Window**
The Phonebook

Pops up the **Revise Phonebook Entry** dialog box for the selected phonebook entry or entries, allowing you to view and edit the entry.

Save [menu command, button]

Where: **Text Editor/File Menu** + **Tool Bar**

See Also: **The Editor**

Writes the current file to disk, overwriting the original version of the file. To rename the file without overwriting the original, use the **Save As** command.

Save As [menu command, button]

Where: **Text Editor/File Menu + Tool Bar**

See Also: The Editor

Pops up a common dialog box, prompting you to enter a name, path and drive to save your text file. Use this command to create a new copy of an existing file without overwriting the original.

Screen [button]

Where: __GIF Viewer/Tool Bar__

See Also: __Zoom__

Enlarges the GIF viewer window to fill the entire screen. Click anywhere on the screen to return to the original window size.

Screen to Capture File [menu command]

Where: **QmodemPro/Edit**

See Also: **Screen to Clipboard**
                      Screen to file
                      Capture
                      Scrollback
                      File Definitions
                      Recording Your Session

Copies the text in the current terminal screen to the default capture file.

Screen to Clipboard [menu command]

Where: __QmodemPro/Edit__

See Also: __Screen to Capture File__
__Screen to File__
__Capture__
__Scrollback__
__File Definitions__
__Recording Your Session__

Copies the text in the current terminal screen to the Windows clipboard, where it can then be pasted into other screens or applications.

Screen to File ... [menu command]

Where: **QmodemPro/Edit**

See Also: **Screen to Capture File**
Screen to Clipboard
Capture
Scrollback
File Definitions
Recording Your Session

Copies the text in the current terminal screen to the default screen trap file.

Screen to Printer [menu command]

Where: **QmodemPro/Edit**

See Also: **Screen to Capture File**
Screen to File
Capture
Scrollback
File Definitions
Recording Your Session

Copies the text in the current terminal screen to the default printer.

Script [button]

Where: **QmodemPro/Tool Bar**

See Also: **SLIQ Script Language Reference**

Pops up a common dialog box prompting for filename, path and drive of the script to execute.

Scrollback [menu command]

Where: **QmodemPro/Terminal**

See Also: **Review**
Scrollback to Capture File
Scrollback to File
Capture
Scrollback
File Definitions
Recording Your Session

As the on-line session proceeds, all text that appears in the terminal window is stored in a special scroll-back buffer. While in this mode, you can scroll backward and forward through previously displayed text.

Scrollback Record [menu command]

Where: **QmodemPro/Terminal**

See Also: **Scrollback to Capture File**
                Scrollback to File
                Capture
                Scrollback
                File Definitions
                Recording Your Session

This command toggles the scrollback buffer **on** and **off.** A check mark appears on the menu next to this item when **Scroll back record** is **on**.

Scrollback to Capture File [menu command]

Where: **QmodemPro/Edit**

See Also: **Scrollback to File**
Capture
Scrollback
File Definitions
Recording Your Session

Copies the text in the current scrollback buffer to the default capture file.

Scrollback to Clipboard [menu command]

Where: **QmodemPro/Edit**

See Also: **Scrollback to File**
Capture
Scrollback
File Definitions
Recording Your Session

Copies the text in the current scrollback buffer to the Windows clipboard, where it can then be pasted into other screens or applications.

Scrollback to File [menu command]

Where: **QmodemPro/Edit**

See Also: **Scrollback to Capture File**
              Scrollback to Printer
              Capture
              Scrollback
              File Definitions
              Recording Your Session

Copies the text in the current scrollback buffer to the default screen trap file.

Scrollback to Printer [menu command]

Where: **QmodemPro/Edit**

See Also: **Scrollback to Capture File**
                Scrollback to File
                Capture
                Scrollback
                File Definitions
                Recording Your Session

Copies the text in the current scrollback buffer to the default printer.

Search Again [menu command]

Where: <u>**File Viewer/Search**</u>
                <u>**Text Editor/Search**</u>

See Also: <u>**The Editor**</u>

Continues the search operation, using the data already in the **Text to find** field in the **Find** dialog box.

Select ... [menu command]

Where: **QmodemPro/Device**

See Also: **Devices**

The **Select** menu pops up a dialog box showing a list of all currently installed devices. Select a device by placing the mouse cursor over the desired device, and click the left mouse button once. Click the **OK** button to confirm your selection and return to the terminal window, or the **Cancel** button to close the window without making changes.

Select All [menu command]

Where: **File Viewer/Edit**
Text Editor/Edit

See Also: **Viewing a File**
The Editor

Selects (highlights) all the text in the current window.

Send ... [menu command]

Where: **QmodemPro/Fax**

See Also: **Phonebook**
Sending and Receiving Fax Messages

Pops up the **Fax Send** dialog box prompting you to enter the device to use, telephone
number to dial, the name of the sender and addressee, the name of the text,
Windows .BMP or .PCX file on disk to send as a Fax, and additional selections for fine or
regular Fax mode, default font, and whether or not to send the cover sheet.

Shell to DOS [menu command]

Where: **QmodemPro/File**

Opens a window which runs the command processor specified in COMSPEC. The **Shell to DOS** command allows you to execute DOS commands or run DOS applications without having to switch back to the Program Manager.

Sort [menu command, button]

Where: **Phonebook/Edit Menu** + **Tool Bar**

See Also: **Sort Phonebook**
              The Phonebook

Pops up the **Sort Phonebook** dialog box, allowing you to rearrange your phonebook file.

Source window

Changes focus from Watch Window to Source Window

Sounds ... [menu command]

Where: **QmodemPro/Options**

See Also: **Customizing the Desktop**

Pops up the **Sounds** dialog box, allowing you to associate Windows .WAV files with certain *QmodemPro for Windows* system events such as connecting and successful download.

Your PC must be equipped with a sound card or a driver for your internal PC speaker, and the sound card or PC speaker drivers must be installed using the Windows Control Panel in order to use this feature.

Split Screen [menu command]

Where: **QmodemPro/Terminal**

In **split screen** mode, *QmodemPro for Windows* divides the screen into two areas to separate incoming and outgoing text. The larger section at the top of the screen is reserved for incoming text received from the host computer. The smaller section at the bottom displays transmitted keystrokes.

Transmitted keystrokes are held in a buffer until the **[Enter]** key is pressed or until you have entered 80 characters on a line.

Step [menu command, button]

Where: **Script Debugger/Debug**

See Also: **SLIQ Script Language Interface for QmodemPro**
             Return
             Jump
             Stop

The Step button causes the debugger to execute one script statement. If the script statement is a subroutine or function call, the debugger will stop on the first statement of the subroutine or function body.

Stop [menu command, button]

Where: **Script Debugger/Debug**

See Also: <u>SLIQ Script Language Interface for QmodemPro</u>
           <u>Return</u>
           <u>Step</u>
           <u>Jump</u>

The Stop button can be pressed at any time during script execution to cause the debugger to stop the script program at the next possible point. Note that the script may not stop immediately, but it will stop when the currently executing script statement has finished.

Stop [button]

When you load multiple files into the GIF viewer, for instance by dragging and dropping from the Windows File Manager, the GIF files are automatically displayed in a slide-show sequence, with a short pause between each file.

The **Stop** button takes the GIF viewer out of slide show mode, and leaves the current image on screen until you select the **next** or **previous** button.

Tab expansion [menu command]

Where: **File Viewer/Options**

See Also **Viewing a File**

When **on**, Tab Expansion expands the tab character (**[Ctrl][I]** or 09 hex) on screen, as white space. When **off,** the tab character is displayed. A check mark appears beside this menu item when it is turned **on**.

Tab Size ... [menu command]

Where: **File Viewer/Options**
Text Editor/Options

See Also: **The File Viewer**
The Text Editor

Pops up a dialog box allowing you to change the number of space characters assigned to a tab.

Tile windows

Arranges the Source and Watch windows side-by-side.

Undo [menu command, button]

Where: **Text Editor/Edit Menu** + **Tool Bar**

See Also: **Redo**
The Text Editor

Removes the results of last editing command you executed. For instance, if you cut or cleared a block of text, the Undo command will restore the text to its original place in the document.

Upload ... [menu command, button]

Where: **QmodemPro/File Menu** + **Tool Bar**

See Also: **Upload File Selection**
Path Definitions
Protocol Options
Uploading and Downloading Files
What to do with Compressed Files
Viewing a File

Uploading refers to copying one or more files from your computer to a host computer such as a BBS or on-line service. The Upload command pops up the **Upload File Selection** dialog box.

View a File ... [menu command]

Where: **QmodemPro/File**

See Also **File Viewer Window**
             Viewing a File

Opens a common dialog box which requests a file name, path and drive, then opens the **File Viewer** window, allowing you to view the contents of a text or binary file. The default paths are filled in from the **Options/Files/Path Definitions** dialog box.

You may substitute your own file viewer in place of the internal one, by specifying the preferred program in the **Options/Files/Executable Programs** dialog box.

View a GIF File... [menu command]

Where: **QmodemPro/File**

See Also: Viewing a File

Opens the **GIF file viewer,** allowing you to view files saved in the CompuServe Graphical Image Format. This is a commonly-used format for storing pictures on your PC.

View Pages ... [menu command]

Where: **Fax Viewer/Page**

See Also: **Fax Viewer Window**

Displays multiple pages of a received Fax message in thumbnail images.

Watch

Where: **Script Debugger/Debug**

See Also: **SLIQ Script Language Interface for QmodemPro**
Return
Step
Jump

The Watch button is used to select a variable to watch in the Watch window. If the cursor in the Source window is positioned over a variable name when the Watch button is pressed, that variable will automatically be watched. If not, a selection box will be presented that will allow you to either pick a variable to watch or type in a variable name.

Watch window

Changes focus from Source Window to Watch Window

Wordwrap [menu command]

Where: **Text Editor/Options**

See Also: **The Text Editor**

When **wordwrap** is turned **on**, text at the right margin automatically wraps to the next line, without requiring a carriage return. When wordwrap is turned **off**, carriage returns must be entered manually.

Wrap at Column ... [menu command]

Where: **Text Editor/Options**

See Also: **The Text Editor**

Pops up the **Wrap Column** dialog box, allowing you to set the right margin.

Xon/Xoff [menu command]

Where: __QmodemPro/Device__

See Also: __Device Connection__

Xon/Xoff, or **software flow control** is a set of signals which start and stop the flow of data. When Xon/Xoff is turned **on**, **[Ctrl][S]** will stop the flow of data and **[Ctrl][Q]** will restart it. Xon/Xoff is generally used to control the flow of text viewed on the screen, or to prevent a data buffer overflow.

A check mark appears beside this menu selection when Xon Xoff is toggled **on**.

Zoom [menu command]

Where: **GIF Viewer/View Menu** + **Tool Bar**
Fax Viewer/Page

Use the Zoom button to enlarge or reduce the section of the image under the mouse pointer. Click the left mouse button to enlarge (zoom in), or click the right button to reduce (zoom out).

Dialog Boxes

This is a complete list of dialog boxes that appear in *QmodemPro for Windows*. Dialog boxes are listed by title. To see help about a specific menu command while youre running *QmodemPro for Windows*, hold down the left mouse button on the menu command youre interested in, and press **F1**. For help with dialog boxes, use the **Help** button on the dialog box, or press **F1**.

ANSI Options

ASCII Options

Desktop Options

Device Connection

Device Hardware

Device Initialization

Device Interface

Device Modem

Device Name

Device Prefixes

Device Select

Device Settings

Dialer

Download Status

Edit Devices [dialog box]

Emulation Color

Emulation Font

Emulation Keyboard Mapping

Emulation Options

Emulation Translation Table

Executable Programs

Fax Options

Fax Receive Log

Fax Send

File Definitions

Find Text (File Viewer)

Find Text (Text Editor)

Find Phonebook Entry

Goto Entry

Goto Line

Open/Edit/Select/View a File

This box allows you to select a file, and browse other directories and drives.

File Name

Type in the file name, if you know it, or use DOS Wildcards * and ? to narrow your search.

Directories

Select a directory to open by double clicking on the directory name. An open file folder icon appears beside the directory listed in the File Name window.

List Files of Type

This window normally lists files that can be opened by the application or menu command which popped up the Common Dialog Box. Click on the arrow to the left of the window to view other selections.

Drives

Click on the arrow on the right side of this window to see other drives.

OK

Accept your selection and proceed with your command, or in the case of a configuration dialog box, return to the previous window, screen or dialog box.

Cancel

Discard your selection and return to the previous window, screen or dialog box.

Help

Pops up a help screen describing the current dialog box or operation.

Select Directory

This dialog box is shown when only a directory selection is needed.

Directories

Select a directory to open by double clicking on the directory name. An open file folder icon appears beside the directory listed in the File Name window. You can also type the desired full directory path in the file window.

Drives

Click on the arrow on the right side of this window to see other drives.

OK

Accept your selection and return to the previous window, screen or dialog box.

Open Dir

Opens the selected directory. Has the same effect as double clicking on the directory name.

Cancel

Discard your selection and return to the previous window, screen or dialog box.

Help

Pops up a help screen describing the current dialog box or operation.

Print

This dialog box allows you to print a text or graphic file or a selection from a text file. This is a standard Windows Print Dialog Box.

Printer

This is the printer *QmodemPro for Windows* will use to print the current document. To change this, go to the **Setup** button on this dialog box.

Print Range

While **Print Range** options are shown on the Print dialog box, these options are not applicable to the Editor or Viewers, so they are grayed out. The **Print** function supports only printing the full document.

Print Quality

The options in this box and the effects they produce depend on the options offered by your printer driver. If this box is dimmed, this option is not available for the selected printer.

Print to File

Default: Off

Save the file to another file that contains commands that can be sent to the printer from DOS. You will be prompted for a file name after you choose the OK button to close the Print dialog box.

Copies

Default: 1

Indicate the number of copies to be printed.

Collate Copies

Default: On

Prints multiple copies in sets, from first page to last page. If this option is turned off, multiple copies of page 1 are printed, followed by multiple copies of page 2, and so on.

Setup ... [button]

Opens **Print Setup** dialog box.

Print Setup

This dialog box allows you to select and configure your printer. The appearance of this dialog box may vary, according to the capabilities of your printer. Refer to your Windows documentation for information on printer setup.

Help Screens

The Windows Help system is extremely comprehensive, and offers an easy way to search for topics that interest you, jump from one topic to another, annotate entries and print help topics. The *QmodemPro for Windows* help system takes full advantage of the Windows help engine.

Command Overview

The Command Overview lists each menu command in order by the menu on which it appears. Click on the text of a topic that interests you for more information.

Contents

The Contents lists all the available help topics, including menu commands, dialog boxes, How-To information and a troubleshooting guide. Click on the text of a topic that interests you for more information.

Index

The Index lists each menu command in alphabetical order. Click on the text of a topic that interests you for more information.

How to Use Help

Pops up the help screen for the Windows help system. This series of screens explains how to use Windows help.

ASCII Options [dialog box]

Upload CR/LF Handling

DOS files with carriage returns and line feeds may cause unexpected behavior on non-DOS computers. When you use ASCII protocol to send a text file to a computer that uses something other than the standard DOS carriage return/line feed way to indicate the end of a line, you may need to add or strip these characters from your document as it is sent.

Send CR

Default: ON

If this option is **on**, Carriage Return characters in your document are as is, without being stripped or translated.

Add LF after CR

Default: OFF

If this option is toggled **on**, a trailing Line Feed is added after each Carriage Return in the document you are sending.

Strip CR

Default: OFF

If this option is toggled **on**, all Carriage Returns are removed (stripped) from the document as it is uploaded.

Send LF

Default: ON

When this option is **on**, Line Feed characters in your document are sent as is, without being stripped or translated.

Add CR before LF

Default: OFF

When this option is toggled **on**, a leading Carriage Return is added before each Line Feed.

Strip LF

Default: OFF

When this option is toggled **on**, all Line Feeds are removed (stripped) from the document as it is uploaded.

Upload Options

Besides carriage return/line feed handling, several other options are available when *QmodemPro for Windows* sends a text file using ASCII protocol.

EOL character

> **Default: 10**
> **(decimal representation of LF**
> **character)**

Otherwise known as End Of Line Character, this is the character that your text editor uses as the last character in a line. Most often, this is a Line Feed character, which is usually preceded by a Carriage Return. The EOL character is used to tell *QmodemPro for Windows* when a line has ended. This information is needed for **Pacing** and **Inter line delay** described below.

If your text editor or word processor uses a different EOL character, perhaps to indicate a soft return, enter its decimal value here. Setting this value incorrectly results in a jerky or incomplete transmission.

Pace character

> **Default: 10**

If a pacing character is specified, and the **Use pace char** box is checked, *QmodemPro for Windows* will wait until that character is received before it sends the next line of text. If this entry is blank, the transfer will not pause between lines of text. Use this only when you are sure that the remote system *always* prompts for each line of text with the same character. To change the setting, enter the decimal ASCII value for the character desired.

As an example, suppose you want to upload a message to your local BBS. The message was created with the *QmodemPro for Windows* text editor. Many BBSs start each message line with a colon ":" (58 decimal), which we can use as the pacing character.

If the BBS also terminates message entry when a blank line is received, be sure to set the Expand Blanks option to **on**.

In some cases, you may want to control transmission pacing yourself. By setting the pacing character to ASCII 126 (a ~ tilde, or any other character which *you know* will not be received by the other system), *QmodemPro for Windows* will pause after sending each line, and wait for you to press the ~ (tilde) key to send the next line. Repeat this procedure until the entire file has been transmitted.

Inter char delay

> **Default: 0**

This is the number of milliseconds to pause between sending characters during an ASCII upload. Try increasing this number if the host system has problems accepting ASCII uploads at full speed.

Inter line delay

> **Default: 0**

This is the number of milliseconds to pause between sending lines during an ASCII upload. Consider increasing this number if you upload prepared text into the message editor on a BBS. The pauses allow the BBS to accept and store each line of data, and return to the editing prompt.

Use Xlate table

> **Default: OFF**

When set to **on**, outgoing data is translated according to the translation tables for the emulation currently in effect.

Use pace character

> **Default: OFF**

When set to **on**, *QmodemPro for Windows* looks for the character entered in the **Pace Character** field as a signal to send a line of text.

Expand blanks

> **Default: OFF**

This option controls whether or not blank lines (lines which contain only a Carriage Return and/or Line Feed and no other text) should be expanded to include at least one SPACE as a leading character.

Note that many BBSs and on-line services interpret a blank line, as defined above, as an end-of-message indication. If your text contains blank lines, the first one will cause the other system to stop receiving message text, issue an end-of-message prompt, and wait for your input. Since additional message text is still being sent, unpredictable results can occur. Setting Blank Line Expansion **on** will eliminate this problem.

Expand tabs

> **Default: OFF**

When set to **on**, a tab character in the original text is translated to 8 spaces. When this option is **off**, the literal tab character (09 decimal) is sent.

Display text

> **Default: OFF**

When set to **on**, text is displayed on the terminal screen during an ASCII upload. When set to **off**, the display of text during an ASCII upload is suppressed, and the upload status window will pop up, displaying the progress of your upload.

Download CR/LF Handling

Line feed handling may be important in how data is received. For example, lines of data sent by a mainframe computer end with a single LF (Line Feed). If you wish to save the incoming data to a file and edit it with an editor or word processor, a carriage return

character should be added in front of each trailing LF and the ASCII download options should be set to turn **off** translation, perform no special Carriage Return handling, and Add CR to Line Feeds.

Receive CR

Default: ON

If this option is **on**, Carriage Return characters in your document are saved as is, without being stripped or translated.

Add LF after CR

Default: OFF

If this option is toggled **on**, a trailing Line Feed is added after each Carriage Return in the document you are receiving.

Strip CR

Default: OFF

If this option is toggled **on**, all Carriage Returns are removed (stripped) from the document as it is downloaded.

Receive LF

Default: ON

When this option is **on**, Line Feed characters in your document are saved as is, without being stripped or translated.

Add CR before LF

Default: OFF

When this option is toggled **on**, a leading Carriage Return is added before each Line Feed.

Strip LF

Default: OFF

When this option is toggled **on**, all Line Feeds are removed (stripped) from the document as it is downloaded.

Download Options

Timeout

Default: 30

Because not all DOS files are terminated with a **[Ctrl][Z]** End of File Marker, *QmodemPro for Windows* needs another way to determine if an ASCII download is complete. The Timeout value is the amount of time *QmodemPro for Windows* will wait after receiving the

last character before it terminates the transfer and saves the file.

Use Xlate table

Default: OFF

When set to **on**, incoming data is translated according to the translation tables for the emulation currently in effect.

Strip 8th bit

Default: OFF

Ignores the high bit, and saves only 7 bit characters. This option might be required if the system or on-line service sends the 8th bit and causes weird looking foreign language characters to appear on your screen instead of regular text.

Display text

Default: OFF

When set to **on**, text is displayed on the terminal screen during an ASCII download. When set to **off**, the display of text during an ASCII download is suppressed, and the download status window will pop up, displaying the progress of your download.

Desktop Options

The desktop pattern and wallpaper are displayed on the area of the screen surrounding the terminal window. The procedure for modifying these settings is the same as the one used in the Windows Control Panel for modifying the Windows desktop.

Pattern

Default: NONE

The **Pattern** option is similar to the one in the Windows Control Panel Desktop dialog box. The selected pattern is displayed in the terminal window, on the screen area outside the borders of the terminal screen.

Foreground
Background

These two list boxes show the colors from which you can make your selection.

Wallpaper

Default: Disabled

Wallpaper is simply a bitmap file created by applications such as Windows Paintbrush. You can place a bitmap file in a repeating pattern, or centered on your screen, where it will be displayed in the areas of the *QmodemPro* window surrounding the terminal screen.

Wallpaper will be displayed on top of any desktop pattern you have selected. If the wallpaper is tiled, the desktop pattern will be covered up.

Filename

Type the name of the .BMP file you want to select, or use the **Edit** button to pop up a dialog box allowing you to select among different files, paths and drives.

Disable

When **on**, no wallpaper is displayed.

Tile

When **on**, the .BMP file is repeated across the screen.

Center

When **on**, the .BMP file is displayed in the center of the screen.

Toolbars

You can display or hide the Tool Bar and Macro bar on your screen.

Toolbar

Default: ON

The Tool Bar contains buttons for frequently-used commands and is displayed at the top of the **QmodemPro** window below the menu bar.

Macro bar

Default: ON

The Macro bar has user-defined buttons displayed at the bottom of the *QmodemPro* window. These buttons correspond with the numeric keys in various shifted, control and alternate combinations.

Terminal Scrolling

Default: Scroll

These two options determine how *QmodemPro for Windows* displays new incoming data on your terminal screen.

Both options have the same net result of showing the newest text at the bottom of the screen, with older text moving up and off the top of the screen, the difference being that depending on your video drivers, one option will be faster and smoother than the other. Trial and error is probably the best method for determining the settings that work best with your system.

Scroll

This option **scrolls** the screen when a new line of data is displayed.

Repaint

This option **repaints** the screen completely when a new line of data is displayed.

Terminal Font

Default: Selected font

This option provides two ways to determine the size of the terminal screen, and the size of the characters displayed inside the terminal screen. Both options are affected by the terminal font you select in the **Emulation Options** dialog box.

Selected font

This is the default setting, which uses the **selected font** and size to determine the size of the terminal window. You must manually select a different font size in **Emulation Options** to change the size of the terminal window.

Based on window

This setting changes the size of the terminal font **based on window** size. You can enlarge or reduce the size of the *QmodemPro* window by dragging the side or corner of the window with your mouse. The fonts in the terminal screen will change size according to the size of the window, if different sizes are available.

Terminal Update

Default: Line

The behavior of this option, like the Terminal Scrolling option, depends on your video drivers, and controls how often *QmodemPro for Windows* updates the screen to display incoming data. If your video hardware is fast enough to keep up with the modem, try **Character** or **Line**. With slower video hardware and drivers, try **Block** for a faster display.

Character

With this option, *QmodemPro for Windows* displays each character as it is received. This is the slowest method.

Line

QmodemPro for Windows waits for a line of text to arrive before updating the screen.

Block

The screen is updated less often than in line mode, and is updated a block at a time. The size of the block depends on several factors, including how much and how fast the data arrives, and what else the CPU is doing at the time.

Accept International Keystrokes

Default: OFF

This option is people to use *QmodemPro for Windows* in a language and character set other than Code Page 437 (North American English). If you are using a different Code Page, and get the wrong characters when you type, try turning this option **on.**

Device Connection

Where: **QmodemPro/Options/Device/Edit Devices/Connection**

See Also: **Hangup**
Xon/Xoff
Break
Full Duplex

Ignore carrier detect

Default: OFF

Some modems always default to reporting Carrier Detected (CD) until told to report the *true* state of carrier. If this is the case, *QmodemPro for Windows* will pop up a dialog box when it starts, asking if you want to send the modem initialization string.

If your modem cannot be set to report the *true* state of carrier detection, turn this option **on**.

In most other cases, however, it is best to leave this option toggled **off**.

Confirm hang-up

Default: ON

When this option is **on**, *QmodemPro for Windows* will pop up a dialog box when you request the hang-up command, confirming that you do in fact want to hang up the modem and break the connection.

Break length

Default: 1000

This is the number of milliseconds to send the BREAK signal, when requested. You should normally leave this value at the default, unless there is a specific reason to change it.

DTR delay

Default: 1000

This is the number of milliseconds to lower the DTR (Data Terminal Ready) signal, when used in conjunction with the Hangup command. Lowering DTR is the standard method of telling the modem to hang up. You should normally leave this value at the default, unless there is a specific reason to change it.

Use of DTR to hang up only becomes active if your settings for **Options/Device/Modem/Hangup** is set to DTR.

Full/Half Duplex

Default: Full

Duplex describes the method used to display your keystrokes on the terminal screen. In Full Duplex mode, the host system echoes your keystrokes back to you. In Half Duplex

mode, the host system does not echo your keystrokes, so *QmodemPro for Windows* must do it instead.

For most connections you can use full duplex, since the remote system will echo your keystrokes.

Device Hardware

Port

Default: COM1 Serial

This is the COM port number associated with the device. Select the port number to which your modem is connected. Select **Serial** if your modem is connected directly to your PC, which is the case for the majority of users.

If you are using a network modem that uses INT 14 connectivity you can select INT 14 without loading any special drivers on the workstation.

If you are using another type of network modem, and if you have already installed the appropriate network modem drivers in Windows, follow the instructions with the network modem drivers for the proper port settings.

Parity

Default: None

Parity is used as a simple means of error checking. Parity values are None, Odd, Even, Mark and Space. This value must match that used by the host system being called.

Most Bulletin Board Systems and On-line Services use None (no parity), the default value. CompuServe uses Even parity.

Stop

Default: 1

Stop bits are generated by the serial port to isolate transmitted characters. The majority of computer systems recognize 1 stop bit, the default value. Supported values are 1 and 2. As with other parameters, this value must match the host computer's setting.

Data

Default: 8

Characters transmitted over modem connections are made up of data bits. Supported values are 7 and 8. For systems which support text and binary program file transfers (that is, most PC-based Bulletin Board Systems), 8 data bits are required. Mainframe and minicomputer-based systems, including CompuServe, commonly use 7 data bits. As with other parameters, this value must match the host computer's setting.

Fax class 1 or 2

If your modem supports CLASS1 or CLASS2 Fax protocol, turn this toggle **on.** This will enable the send and receive Fax functions in *QmodemPro for Windows*. If your modem does not support sending or receiving Fax messages, leave this option turned **off**.

Baud rate

> **Default: 2400**

The baud rate is the speed at which the COM port operates and should in most cases match the top speed of the device, or the locked DTE speed. High speed modems, that is, modems with nominal baud rates of 9600, 14400, 16800 and higher, should always be set with the DTE locked to a higher baud rate than the actual connect speed, to enable hardware flow control, data compression and error correction. Thus, the correct setting for a 9600 or 14400 baud modem would be 19200 or higher.

Flow Control

There are several types of flow control settings used by *QmodemPro for Windows.*

Xon/Xoff

> **Default: OFF**

Xon/Xoff is a software flow control protocol used during normal text and ASCII transmissions. It uses **[Ctrl][S]** and **[Ctrl][Q]** characters to stop and start the flow of data.

Mainframe systems can use Xon/Xoff as a way to pace *text* data transmissions. Most BBS systems do not use Xon/Xoff.

If the Xon/Xoff switch is **on**, *QmodemPro for Windows* will send the Xoff character when its receive buffer is almost full.

As text is displayed, the receive buffer empties and can accept more data. *QmodemPro for Windows* then sends the Xon character to resume transmission.

CTS/RTS

> **Default: ON**

CTS/RTS is known as hardware handshaking. It makes use of the CTS (Clear To Send) and RTS (Request To Send) signals on your serial port.

This is a faster and more reliable method of flow control because it is performed by the modem hardware, and is the recommended setting for high speed and error-correcting modems.

In order for CTS/RTS flow control to function properly, the serial cable (if you are using an external modem) must support the CTS and RTS signal lines.

DSR/DTR

> **Default: OFF**

DSR/DTR flow control is similar to CTS/RTS, but uses the Data Set Ready and Data Terminal Ready signal lines.

This flow control setting is only used in special situations which require it, and should normally be switched **off**.

DTE Locked

> **Default: ON**

The DTE (Data Terminal Equipment) speed, or port speed, is the rate at which your computer sends data to the serial port, as opposed to the speed your modem sends data to the other modem.

High speed modems work best when the DTE rate is locked at a higher speed than the actual connect speed. This allows the modems themselves to negotiate the fastest connection between them. For instance, a 9600 or 14400 modem should have the Baud rate set to 19200 or 38400. Some modems are capable of higher locked baud rates, but the maximum practical speed depends on the speed of your CPU.

When both the sending and receiving modems lock the DTE rate and use a streaming transfer protocol such as Zmodem, the characters per second performance can actually *exceed* the nominal baud rate of the modems.

If DTE is not locked, the DTE connection matches the modem connect speed. To operate properly, your modem must also be told to lock the DTE in the initialization string.

So, for best performance with a high speed modem, leave this toggle **on**.

Fax Baud rate

Default: 9600

This is the maximum Fax baud rate supported by your Fax modem. *QmodemPro for Windows* will automatically negotiate a lower rate upon connection, if necessary.

Allowable values are 2400, 4800, 7200, 9600, 12000 and 14400. Consult your modem manual for the correct setting here.

Device Modem

This window allows you to define the command strings and result codes used by *QmodemPro for Windows.*

Modem initialization

> **Default: ATZ^M**

This is the **initialization string** sent to the modem to restore it to its standard operating parameters. It is sent when *QmodemPro for Windows* is started, and when the menu choice **Device/Reset Modem** is selected. The command in this window may vary depending on the modem being used; consult your modem manual for an explanation of each command in the initialization string. The ^M (**[Shift][6]** + **[M]**) is translated to a carriage return or **[Ctrl][M]**.

Dial prefix

> **Default: ATDT**

This command is used to tone-dial a number on a Hayes compatible modem. This signals the modem to dial the numbers sent immediately after this command. If your phone system requires pulse dialing, change the dial prefix to ATDP.

The dialing commands may vary depending on the kind of modem. Consult the modem manual for additional information. If your modem requires a 150 millisecond BREAK signal for dialing, change this command to the word BREAK.

Adding *70 after the ATDT dial command disables call-waiting on some tone-dialing phone systems. On pulse-dialing phone systems try adding 1170 after the ATDP command. Check with your local phone company to see if these features are supported.

Dial suffix

> **Default: ^M**

This string immediately follows the number to be dialed. The normal suffix is a carriage return **[Ctrl][M]**, entered as ^ (**[Shift][6]**) M.

See **Special Control Characters** for more information on how to enter control code sequences in macros, modem command strings, and scripts.

Connect Message

> **Default: CONNECT**

This message is returned by the modem when it establishes a successful connection. Most modems will also report the connect speed as part of the message, for instance:

```
CONNECT 19200
```

Do *not* include a speed value as part of this message since it will vary.

If you are making a direct connection to another PC without the use of modems, enter the word DIRECT in this field. See **Direct connection** for more detailed information.

No connect 1 - 4

Defaults:	BUSY
	NO CARRIER
	NO DIAL
	VOICE

These messages are reported by the modem when it could not establish a connection. These messages are used to tell *QmodemPro for Windows* that a connection was not made, and that it should recycle and dial again. Your modem may return messages slightly different from the examples shown above; refer to your modem manual for exact wording.

OK message

Default: OK

The modem returns an "OK" message when the last command was received and executed. For Hayes compatible modems, use OK. For modems which do not return a response to the initialization command, leave this field blank.

Cancel dial

Default: ^M^M

This string will be sent to the modem when you cancel a dial request. If the modem supports lowering DTR (Data Terminal Ready), to disconnect, enter the letters DTR in this field. Consult your modem manual for information on this feature.

Answer on

Default: ATS0=1^M

This command puts the modem into Auto Answer mode, and is sent when you select the **Device/Auto Answer ON** menu command. The modem will answer the phone on the first ring, and will try to negotiate a connection.

Answer off

Default: ATS0=0^M

This command takes the modem out of Auto Answer mode, and is sent when you select the **Device/Auto Answer OFF** menu command. The modem will not try to answer the phone when it rings.

Hangup

Default: DTR

Most modems recognize the lowering of the DTR (Data Terminal Ready) signal as a disconnect method. To enable this method, enter the letters DTR in this field. This

hardware-only method is far more reliable than using a modem software command, which could easily fail because of timing requirements or line noise.

The modem command entered here is sent to the modem when **Device/Hangup** or the **Hangup** button are selected.

If your modem does not support the lowering of DTR to hang up, try entering

```
^~^~^~+++^~^~^~ATH0^M
```

This command will cause Hayes compatible modems to hang up most of the time. Each ^~ sequence creates a 1/2 second delay, and the +++ sequence means "escape and go into command mode". The command ATH0 tells the modem to go back on hook, and the ^M ends the command string. Enter the word BREAK in this field if your modem or switchboard requires a 150 millisecond BREAK signal to disconnect a call.

See **Special Control Characters** for complete information on how to enter control code sequences in macros, modem command strings, and scripts.

Redial interrupt

```
Default: RING
```

If *QmodemPro for Windows* receives this string from the modem during dialing, it will stop dialing immediately. If you are sharing a voice line with your modem, this allows you to answer the incoming phone call. If your modem is connected to its own dedicated phone line, you can leave this field blank.

Delay during

```
Default: 60
```

This is the number of seconds *QmodemPro for Windows* waits for a connection after dialing a number. When this period has elapsed, *QmodemPro for Windows* will hang up the modem, wait, then dial the next number, if any, or redial the same number. The default value is 60 seconds.

Most modems also feature their own internal time-out delay. When dialing, the shortest of these two values -- hardware or software -- will prevail, so make sure that your modem's initial time-out value is at least equal to the *QmodemPro for Windows* setting. In Hayes compatible modems, the command ATS7=60 will set the modem's internal delay to 60 seconds.

Delay after

```
Default: 10
```

This is the number of seconds *QmodemPro for Windows* waits after an unsuccessful dial attempt (that is, a BUSY signal, expired dial time or manual cycle), before trying the next number in the dialing queue. The default value is 10 seconds. This is also referred to as the **cycle** time.

Device Name

Where: **QmodemPro/Options/Device/Edit Devices/Name**

See Also: **Device**
 Fine Tune
 Devices
 Other Communication Devices

This dialog box allows you to change the name of a device. In most cases it's a good idea to use a device name which describes the modem, and any special settings you may have applied to this device configuration.

Device Prefixes

Where: **QmodemPro/Options/Device/Edit Devices/Prefixes**

See Also: **Phonebook**
Revise Phonebook Entry
Manual Dialer

QmodemPro for Windows allows up to 10 prefix codes per device, assigned to the letters A through J. Each prefix code can store up to 30 characters. You can use multiple prefix codes in any position in the number field, although you should keep in mind that most modems can take a maximum of only 40 characters in a single command string.

Device Select

This procedure will automatically program your modem with the settings that work best with *QmodemPro for Windows*.

Select a device to install

Select your modem from a list of pre-configured definitions. If you cant find your modem on this list, choose one of the Generic modems which most closely matches your own modems baud rate and features.

Load

QmodemPro for Windows stores modem configuration information in a Device Definition Table or DDT file. From time to time, Mustang Software, Inc. updates these files and makes them available for download from the MSI HQ BBS and other online services.

To select a different DDT file, click the **LOAD** button.

OK

Click the **OK** button when you have selected your modem from the list.

IMPORTANT NOTE: *QmodemPro* will search your computers hardware to find available com ports. This procedure may take some time, occasionally more than a minute as it talks to each port it finds. Your computer may appear hung during this process -- it will resume the configuration procedure in a minute or so.

Device Settings

Where: **QmodemPro/Device/Fine Tune**

See Also: **Device Hardware**
Devices

This dialog box allows you to **fine tune** your current device settings. Changes take effect for the current session only. If you change devices, your settings will revert to the ones associated with the new device.

If you need to use this option frequently, consider making these alternate port settings available as an additional device. Go to the **Options/Device** menu, and **Duplicate** your current device, making the necessary changes there. It's then a simple matter to associate a device with the alternate port settings, with a Phonebook entry.

For an explanation of how to use the **Baud Rate, Parity, Stop** and **Data** entries, see **Device Hardware**.

Device Interface

Default: COM1 Serial

Interface

Serial (Windows)

Select **Serial** if your modem is connected directly to your PC, which is the case for the majority of users.

INT14 / EBIOS

If you are using a network modem that uses INT 14 connectivity you can select INT 14 without loading any special drivers on the workstation.

If you are using another type of network modem, and if you have already installed the appropriate network modem drivers in Windows, follow the instructions with the network modem drivers for the proper port settings.

Port

This is the COM port number associated with the device. Select the port number to which your modem is connected.

IMPORTANT NOTE: *QmodemPro* will search your computers hardware to find available com ports. This procedure may take some time, occasionally more than a minute as it talks to each port it finds. Your computer may appear hung during this process -- it will resume the configuration procedure in a minute or so.

Recommended Switch Settings

Where: **Device Select**

See Also: Edit Devices

This window contains technical notes on making the best use of your modem. Follow the instructions on switch settings and other technical notes carefully

Device Initialization

Where: **Device Select**

See Also: Edit Devices

This dialog box prepares to send commands which will program your modem for best performance and operation for *QmodemPro for Windows*.

Dialer

The Dialer dialog box displays the current status of the call being dialed.

Dialing List

The **dialing list** window shows all entries in the current dialing queue. The entry currently being dialed is highlighted in a contrasting color.

Details

The **Details** window shows information about the entry currently being dialed. This information is extracted from the Phonebook entry for that particular system.

Status Messages

The **Status Messages** window shows a list of commands sent to the modem, responses from the modem, and connect messages. When a successful connection is established, the Dialer dialog box is replaced by the Terminal Screen, or the Fax Status dialog box, depending on the type of connection made.

Progress

The **Progress** window shows the number of seconds to wait for a connection, the seconds between dialing attempts, the current time, and the start time for the current dialing attempt.

Cycle

The **Cycle** button aborts the current dialing attempt and goes on to the next entry in the dialing queue. If there is only one entry in the dialing queue, that entry will be redialed after pausing for the number of seconds shown in the **Seconds between** field in the Progress window.

Untag

The **Untag** button removes the highlighted entry from the dialing queue. Any entry, active or not, can be selected.

Extend

Use the **Extend** button to extend the length of time *QmodemPro for Windows* will wait for a successful connection with the current entry. Use this if the **Wait for connect** counter is nearing zero and the system has not answered the phone or the modems have not yet established a connection.

Note that modems usually have an internal timer that may override the time extension. Refer to your modem manual for instructions on changing this internal setting.

Cancel

The **Cancel** button aborts the current dialing attempt and closes the Dialer dialog box.

Download Status

Protocol

Displays the current file transfer protocol.

Block check

This indicates the error correction mode: Checksum, CRC 16, CRC 24, or CRC 32, depending on the transfer protocol in use.

File name

The name of the file currently being received.

File size

The size in bytes of the file currently being received. Some transfer protocols, such as Xmodem, do not send this information, so this field may be blanked out.

Block size

The size of the block currently being received. Block size is dependent on transfer protocol. For instance, Xmodem CRC transmits files in 128 byte blocks, while Ymodem transmits 1 kilobyte (1024 byte) blocks. Zmodem adjusts the block size depending on line conditions, but defaults to 1 kilobyte blocks.

Total blocks

The total number of blocks in the file currently being received. Some transfer protocols do not send this information, so this field may be blanked out.

Bytes transferred

The number of bytes already received in the current file.

Bytes remaining

The number of bytes remaining to be received in the current file, if known -- depending on the transfer protocol.

Blocks transferred

The number of blocks already received in the current file.

Blocks remaining

The number of blocks remaining to be received in the current file, if known.

Block corrections

The number of times the current block has been retransmitted due to errors detected during the transfer.

Total corrections

The total number of block corrections for the current transfer.

Est. time

The total estimated time to receive the current file, if file size is provided by the transfer protocol. Estimated time is calculated from the reported connect speed, size of the file, elapsed time for the current transfer, and the remaining number of bytes and blocks to be received. The estimated time is recalculated periodically based on the current transfer performance.

Elapsed time

The total elapsed time since the beginning of the current file transfer, in hours, minutes and seconds.

Remaining time

The estimated time to receive the remainder of the current file, if the file size is provided by the transfer protocol. Remaining time is calculated from the reported connect speed, size of the file, elapsed time for the current transfer, and the remaining number of bytes and blocks to be received. The remaining time is recalculated periodically.

Throughput

The number of characters per second received during the current file transfer. This value is recalculated periodically, and represents an average throughput for the file currently being received.

Efficiency

Efficiency is calculated in percent, based on a calculation of the connect speed reported by your modem and the average characters per second.

Status

This window displays any messages generated by the protocol before, during and after the transfer. Most of these messages are for notification purposes only, and do not necessarily indicate an error.

Progress %

This is a graphical thermometer style display, showing a visual representation of the percentage of the file which has already been received, if file size information is provided by the transfer protocol.

Cancel

This control button cancels or aborts the current file transfer. *QmodemPro for Windows* will stop receiving the current file and return to the terminal window.

Edit Devices

Where: **QmodemPro/Options/Device**

See Also: **Device Name**
          Device Hardware
          Device Modem
          Device Prefixes
          Device Connection
          Device Select
          Fine Tune

This dialog box lists the devices you have configured, and the current device, and allows you to add, edit and delete devices from the list.

Devices

This is a list box of available devices. Up to 10 separate devices can be created. To select a device for editing, move the mouse pointer to the device you want to select, and click the left mouse button.

Default Device

This window shows the name of the default device. To select a device as your default, highlight it in the Device list, and click the **Set Default** button. The default device is activated at startup.

Name

This is the name assigned to the device. To change the name, click the **Name** button and enter the new name in the text entry window that pops up.

Hardware ...

This button pops up the **Device Hardware** dialog box which allows you to assign baud rate, parity, port ID and flow control to a device.

Modem ...

This button pops up the **Device Modem** dialog box allowing you to enter modem commands and result strings.

Prefixes ...

This button pops up the **Device Prefixes** dialog box allowing you to enter prefixes for modem dialing commands.

Connection ...

This pops up the **Device Connection**s dialog box controlling a variety of settings which affect an on-line connection.

Quick Config ...

This pops up the **Device Select** dialog box, allowing you to add and configure a modem as a device in *QmodemPro for Windows.*

Duplicate

Copies the selected device and appends the copy to the end of your Device list. This is useful in cases where you want to use the same device, but with different initialization or dial strings, or parity settings, to accommodate different host systems you call.

Remove

Deletes the highlighted device.

Set Default

Sets the highlighted device as the default device for terminal window commands and new Phonebook entries.

Close

Closes the dialog box and saves your new device settings.

Edit Group

The **Edit Group** dialog allows you to define and assemble multiple phonebook entries into a group.

Group Name

Use the **Group Name** field to store the name of the group, using a descriptive name such as Mailrun or Games or Local BBSs or anything else that describes what these systems have in common.

Dial Mode

Default: OFF

Check the **Continuous Dial** box to dial each entry until you have connected with each system on the list. If this box is not checked, the dial session will stop after the first connection.

Phonebook Entries

The **Phonebook Entries** list shows all the entries in the current Phonebook. To select an entry, highlight it with a click of the left mouse button, then click the **Add** button, or double-click the entry..

Selected for Group

The **Selected for Group** list shows the Phonebook entries belonging to this group. To remove an entry, highlight it with the left mouse button, then click the **Remove** button. You can also double-click an entry to remove it.

Add

Adds the selected entry or entries in the Phonebook Entries list to the Selected for Group list.

Remove

Removes the selected entry or entries from the Selected for Group list.

Clear

Removes all entries from the Selected for Group list.

Edit Groups

Where: <u>**Phonebook/Edit/Groups**</u>

See Also: <u>**Edit Group**</u>
<u>**Phonebook**</u>
<u>**The Phonebook**</u>

This dialog box allows you to create and maintain Groups of Phonebook entries, and mark members of the selected group for dialing and other operations.

Add

Pops up the **Edit Group** dialog box, allowing you to create a new Group.

Revise

Pops up the **Edit Group** dialog box, with information from the selected Group, ready for editing.

Delete

Deletes the selected group or groups.

Tag

Tags all phonebook entries matching the list of entries in the selected group and closes the dialog box. From there, you can dial, revise, or delete the tagged Phonebook entries.

Dial

Immediately dials all entries in the selected group. The **Dialer** status window pops up, showing the progress of the queued calls.

Emulation Color

This group of options changes the color of foreground and background text in the Terminal screen. To change the color and pattern on the Desktop, go to the **Options/Desktop** menu.

Foreground

Default: White

This is the **foreground** color. The letters you see on your screen will be displayed in the selected color. To change the color, click the arrow on the right hand side of the list box and click on your selection.

Background

Default: Black

This is the **background** color, or the color of the screen behind the letters. To change the color, click the arrow on the right hand side of the list box, then click on your selection.

Sample text

The **Sample Text** view shows the foreground and background colors you have selected.

Emulation Font

This dialog box allows you to change the Font and Size of the text displayed in the Terminal Screen. Note that only monospaced fonts are available for selection, including monospaced TrueType fonts.

Font

Default: Terminal

This is the name of the font to be displayed in the Terminal Screen.

Size

Default: 8 x 14

This is the size in pixels (height by width) of the font in the Terminal Screen. Larger numbers mean larger characters. You can see your results in the Preview and Sample windows.

Preview

The **Preview** window shows a sample of the dimensions of the terminal screen, relative to the desktop in the *QmodemPro* window. If the terminal screen will not fit in the desktop window with the selected font, arrows indicate that the screen actually extends beyond the limits of the *QmodemPro* desktop. Arrows may indicate the terminal screen is too large horizontally, vertically or in both directions.

Sample

The **Sample** window shows the actual size and style of the typeface you've selected.

Emulation Keyboard Mapping

Where: **QmodemPro/Options/Emulation Options**/Keyboard

See Also: **Key Mapping**
Keyboard Options
Emulation Options
Macro Keys
Programmable Keys

This dialog box allows you to add and change function key assignments, and key mappings. It shows a graphical representation of the keyboard, with certain keys selectable while others are grayed out and are not selectable. These selectable keys can be assigned text and internal *QmodemPro for Windows* commands.

To assign a command to a key, click on the key you want to change. The **Key Mapping** dialog box will pop up prompting you enter the text for the normal, shift, control and control+shift positions for the key.

Options ...

The **Options** button pops up the **Keyboard Mapping Options** dialog box.

Emulation Options

Where: **QmodemPro/Options/Emulations**

See Also: **Phonebook**
Emulation
Select Font
Translation
Emulation Color
Emulation Keyboard Mapping
RIPscrip Options
Changing the appearance of the terminal screen
Emulations
Programmable Keys

The Terminal Emulation feature allows *QmodemPro for Windows* to behave like a terminal other than a standard PC display, for example the VT100 from Digital Equipment Corp. While in terminal emulation mode, *QmodemPro* interprets keyboard and video information so that it is displayed properly on the PC screen as if it were a native terminal of the specified type. Note that terminal emulations can also be specified for individual phonebook entries.

Emulation

This is a list box which you can open by clicking on the arrow symbol. Select the emulation you wish to edit by positioning the mouse cursor over the desired item, then click the left mouse button.

The rest of the options in this dialog box are applied *only* to the selected emulation.

Toggles

Strict color

Default: OFF

In ANSI and VT100 emulation, there is a question on how the default colors should be set. Both ANSI and VT100 use the Esc[m and Esc[0m command to reset to a color. By the books, this color is white on black. In the past, these Esc codes were set to whatever the default Screen Text color was, as set in the color configuration section. But, depending on the system called, this may result in an incorrect display of either ANSI or VT100 if the designer relied on the white on black colors.

Strict Color will control what color is selected when Esc[m or Esc[0m is executed. When set **on**, the color becomes white on black. When set **off**, it becomes the color set by **Emulation color** under **Emulation Options/Color**. To remain compatible with previous versions of *QmodemPro*, the default is **off**.

Line wrap

Default: ON

This controls how the cursor reacts when it writes a character in the last column on the screen (usually column 80). When set **on***,* the cursor moves the next line down, performing a Line Feed if the cursor was on the last line on the screen. When set **off***,* the cursor writes the character in the last column, but does *not* advance to the next line.

Display nulls

```
Default: OFF
```

This controls how the NULL (0x00) character is handled when received. If this is set **on**, then the Null character is displayed as a Space. When set **off**, the NULL is stripped from the incoming screen.

Because the Translation tables allow stripping characters by changing them to NULL, this only affects the NULL character before translation. Any other character set to a NULL by input translation will be stripped and not be displayed. The only time you might want this to be set **on** is if you wanted to use the ASCII download to capture ALL incoming characters to a file.

Destructive backspace

```
Default: OFF
```

This controls whether or not the character to the left of the cursor is erased when a Backspace character (^H) is received. When **on**, the character is erased before the cursor is moved to the left. When **off**, the cursor moves leaving the character to the left untouched.

Add LF to CR

```
Default: OFF
```

Different kinds of computers use different characters to indicate a line ending. The IBM PC and compatible machines use two characters, a Carriage Return and a Line Feed, to end a line and move to the beginning of the next line. Some other computers, however, use only a Carriage Return, with no line feed.

If each line of text in terminal overwrites the previous line, rather than displaying on the next line of the screen, try setting this to **on**.

Strip high bit

```
Default: OFF
```

This option allows you to connect to a 7-bit system or service with 8-bits selected. Normally when you connect to a 7-bit system, the high bit is set (turned **on**) by the hardware on the receiving end. The high bit is then interpreted by the software as a meaningful bit when it really is not meaningful.

This toggle turns the stripping of the high-bit **on** and **off**. *QmodemPro for Windows* starts in 8-bit mode by default. This can be changed with a Script command to automate high-bit stripping, for example when calling CompuServe. Some BBSs use fancy graphics and symbol characters. To display these properly, 8 bit mode *must* be in effect.

Default emulation

Displays the default emulation. Use the **Set Default** button to make the highlighted emulation the default.

Terminal Size

Screen height

> **Default: 25**

Change this value to make your screen taller or shorter.

Screen width

> **Default: 80**

Select 132 columns, if the remote host supports this mode, to make your screen wider.

Enquire

Some host systems send an **Enquiry** (ENQ or ^E) sequence to determine the terminal type. These options allow you to determine whether to respond to an ENQ from the host and what your system's response should be.

Disable/Enable/CIS B+

> **Default: CIS B+**

The **Enquire Sequence** is used in various ways, most commonly to start a CIS B+ download automatically. Select **Enable** to allow *QmodemPro for Windows* to send the **Enquire Sequence** or Answerback in response to an ENQ signal.

Enquire Sequence

> **Default: BLANK**

Fill in this field with the Enquire Sequence expected by the host. If you are unsure whether or what to send, consult the system administrators for the host you will be calling.

Set Default

Allows you to select the default emulation for the terminal window and new phonebook entries. This is the emulation that takes effect when you start *QmodemPro for Windows.* Use the **Terminal/Emulations** menu to change terminal emulations during a session.

Translate ...

Opens the **Emulation Translation Table** dialog box, allowing you to view and edit incoming and outgoing character translations. Translations are stored separately for *each* emulation.

Font ...

Opens the **Emulation Font** dialog box, allowing you to view and change the style and size of the characters displayed in the Terminal Window. Note that only monospaced fonts are available for selection.

Color ...

Opens the **Emulation Color** dialog box, allowing you to change the foreground and background color of text in the Terminal Window.

Keyboard ...

Opens the **Emulation Keyboard Mapping** dialog box, allowing you to attach commands to function keys and other keys on your keyboard. Emulations such as VT100, ANSI, and others have default key assignments which you can modify from here.

Options ... [RIPscrip emulation only]

Opens the **RIPscrip Options** dialog box, allowing you to select TrueType fonts and flood fill modes for RIPscrip emulation only.

Emulation Translation Table

Translation may be necessary when communicating with another computer whose ASCII codes (the computer's alphabet) do not match those used by your PC. There are 256 ASCII characters, ranging in value from 0 to 255.

To change (translate) an incoming or outgoing ASCII value, double-click on the character you wish to change, and enter the new value in the dialog box which pops up.

Incoming Translation
Outgoing Translation

The Incoming Translation table controls which incoming characters are passed unchanged and which are translated to other characters, while the Outgoing Translation table controls which characters are sent out your modem to the other system.

Incoming or Outgoing tables are list boxes containing all 256 characters in the ASCII character set from 0 through 255 and their respective translations.

Edit Format

Default: Character

The display mode depends on which option in the **Edit Format** box is checked: ASCII, Decimal or Hexadecimal. In ASCII format, the characters are displayed as text. In Decimal format, characters are represented by their ASCII character number in decimal. In Hex format, the characters are represented by their ASCII character numbers in Hexadecimal (base 16) format.

Some characters cannot be entered from the keyboard, but can be entered as their decimal or hex equivalents.

Executable Programs

This selection allows you to define your choices for internal or external File Viewer and Text Editor. In all cases, you can use the **Edit** button to pop up a common dialog box allowing you to browse file names, paths and drives for your selection.

Viewer Options

Viewer

You have the option to replace the internal viewer with an external viewer of your own, if you prefer. Only file viewers for Windows can be used here.

Viewer Path

This is the default Path passed to the viewer program, where it should look for files to load. Depending on how the viewer is invoked, a filename may be appended to this path. This is *not* where your external viewer program is located.

Internal (path only)

Selecting Internal (Path Only) will use the Internal Viewer. The (Path Only) means that only the Path specified will be used, and the external program specified will be ignored. Selecting External will use the external viewer specified and the above path as the default.

External

Select this option to activate the external viewer defined in the Viewer text box.

Editor Options

Editor

You have the option to replace the internal editor with an external editor of your own, if you prefer. Editors for both Windows and DOS can be used here.

Editor Path

This is the default Path passed to the editor program, where it should look for files to load. Depending on how the editor is invoked, a filename may be appended to this path. This is *not* where your external editor program is located.

Internal (path only)

Selecting Internal (Path Only) will use the Internal Editor. The (Path Only) means that only the Path specified will be used, and the external program specified will be ignored. Selecting External will use the external editor specified and the above path as the default.

External

Select this option to activate the external editor defined in the Editor text box.

Fax Options

Fax number

This is your Fax telephone number. This is placed at the top of each page in small print, and is also extracted from the @FROMPHONE@ Fax Variable for use in a custom cover sheet. Leave this blank if you do not want to display your phone number.

From or Company Name

Put your company name or other identifying text here. This is placed at the top of each page in small print, and is also extracted from the @FROM@ Fax Variable for use in a custom cover sheet. Leave this blank if you do not want to display your name.

Cover sheet filename

Default: C:\qmwin\cover.sht

This is the name and path of the ASCII file you want to send as your cover sheet. Use the **Edit** button to browse and select from other drives, paths and file names. See the examples in Chapter 3 of your User Guide for creating a custom cover sheet.

Fax receive path

Default: C:\qmwin\fax

This is the directory into which *QmodemPro for Windows* will place incoming Fax messages. Fax messages are stored as files which you can view and print from the Fax Viewer window. Use the **Edit** button to browse and select from other drives and paths.

Resolution Normal/Fine

Default: Normal

Two resolution modes are supported, Normal and Fine. They correspond to the "Normal" (200x100 dpi) and "Fine" (200x200 dpi) settings on your Fax device. Fine resolution looks better, but takes longer to transmit.

Fax Receive Log

The Fax Receive Log lists incoming Fax messages, and allows you to view, print, or delete messages.

Date

This is the date the Fax message was received by *QmodemPro for Windows,* and is included in the header information in every Fax message.

Time

This is the time the Fax message was received by *QmodemPro for Windows,* and is included in the header information in every Fax message.

Pages

This is the number of pages in the Fax message received, and is included in the header information in every Fax message.

Identifier

This is normally the telephone number of the Fax machine which sent the Fax message.

View ...

This button opens the **Fax Viewer** window, displaying the selected Fax on screen in a variety of modes.

Print ...

This button prints the Fax highlighted in the Receive Log window.

Delete

Use the **Delete** button to delete the highlighted Fax message from your disk.

Auto Receive

Default: OFF

Turn this option **on** if you would like *QmodemPro for Windows* to receive incoming Fax messages automatically. When this box is checked, the phone will be answered in Fax Receive mode and the incoming Fax documents will be stored on your disk for review in the **Fax Viewer.** In order for Auto Receive to remain active, the **Fax Receive Log** dialog must remain open on screen. Closing the log stops Auto Receive.

Fax Send

Where: **Phonebook/Dial** **[for entries tagged as "Fax Only"]**

See Also: **Phonebook**
　　　　　　Sending and Receiving Fax Messages

This dialog box pops up when you dial a Phonebook entry marked "Fax Device Only", and prompts you for the information *QmodemPro for Windows* needs to send a Fax message from the Phonebook.

Fax Information

The following information is used to fill in Fax Variables in the cover sheet. For more information on Fax Variables, see Chapter 3 of your User Guide.

To

The name of the person to whom the Fax is addressed. This is placed in the @TO@ Fax Variable.

From

Your name, or the name you want the Fax to be from. This is placed in the @FROM@ Fax Variable.

Subject

The subject of the Fax message. This is placed in the @SUBJECT@ Fax Variable.

Document

The name and path of the document to send. Use the **Edit** button to browse other filenames, paths and drives. Documents can be plain ASCII text, .PCX files or Windows Bitmap .BMP files.

Options

Use fine mode

Default: OFF

Two resolution modes are supported, Normal (200 x 100 dpi) and Fine (200 x 200 dpi). settings for your Fax device. Faxes sent in **fine mode** take longer to send.

Include cover page

Default: OFF

Turn this option **on** to send the cover page defined in **Options/Fax** before the selected document.

Font

Pops up the **Select Font** dialog box, allowing you to select the type style and size to use to send the text in your Fax messages. In contrast to the terminal window fonts, both

monospaced (fixed-pitch) and proportional (variable-pitch) fonts can be selected here.

File Definitions

Screen Trap File

Default: SCREEN.TRP

This is the filename assigned to the screen trap or snapshot file. When you request a Screen Trap from the menu command **Edit/Screen to File**, you have the option to accept or change this path on a temporary basis before the file is saved.

Capture File

Default: CAPTURE.CAP

This is the filename assigned to the capture file. When you open a Capture File using the menu command **File/Capture** or by clicking the **Capture** button, you have the option to accept or change this path on a temporary basis before the file is saved.

Log File

Default: QMPRO.LOG

This is the filename assigned to the session log. When you open the Log File from the menu command **File/Log Toggle**, you have the option to accept or change this path on a temporary basis before the file is saved. You can start the log file automatically for each session by selecting **Auto Start Log** on the **Options/File Definitions** dialog box. See the Appendix at the end of this book for a list of Log File entries.

Scrollback File

Default: SCROLL.SAV

This is the filename assigned to the scrollback buffer when it is copied to disk. The scrollback buffer stores incoming data for review. When you save the Scrollback buffer, you have the option to accept or change this path on a temporary basis before the file is saved.

Print File

Default: PRN

This the default print device. This can be a printer port, such as PRN, LPT1, or LPT2, or it can be a file on disk, allowing printer output to be redirected to a file. The default PRN sends all printer output to the printer at LPT1. The Windows printer drivers will automatically format the print output for your selected printer.

Autostart Log

Default: OFF

When this option is **on**, *QmodemPro for Windows* automatically records activity in the Log file (defined above, under **Options/Files**). When this option is **off**, the Log file can be turned on manually from the File menu.

Date/Time Stamp Log

Default: ON

When this option is **on**, the Screen Trap, Capture and Scrollback files have time and date stamp headers placed before the appended data. Note that this time stamp appears when the file is opened, and in the case of the Log file, with each line added to the file.

Find Phonebook Entry

Searches for text in the Phonebook. All Phonebook entry fields are searched, including the text in a Note file attached to an entry (but not the contents of a script file), and the search is case insensitive.

Text to find

Enter the text you want to find in this window, up to a maximum of 27 characters.

Direction

Forward

Default: ON

Searches forward from the cursor, towards the end of the file.

Backward

Default: OFF

Searches backwards from the cursor, towards the beginning of the file.

Find Next

Continues the search operation, using the data already in the **Text to find** window.

Tag All

Tags all Phonebook entries with text matching the **Text to Find**.

Find Text (File Viewer)

Searches for text in the current file loaded in the File Viewer window.

Text to find

Enter the text you want to find in this window, up to a maximum of 80 characters.

Find options

Forward

Default: ON

Searches forward from the cursor, towards the end of the file.

Backward

Default: OFF

Searches backwards from the cursor, towards the beginning of the file.

Case sensitive

Default: OFF

When toggled **on**, finds only text which matches the capitalization of the search text exactly.

Find Next

Continues the search operation, using the data already in the **Text to find** window.

Find Text (Text Editor)

Where: __Text Editor/Search/Find__

See Also: __Replace__
__The Text Editor__

Searches for text in the Text Editor window.

Text to find

Enter the text you want to find in this window, up to a maximum of 80 characters.

Search options

Case sensitive

Default: OFF

When toggled **on**, finds only text which matches the capitalization of the search text exactly.

Whole words only

Default: OFF

When toggled **on,** the search finds only complete words matching the search text. For instance, if you are searching for "fish" in text that reads:

```
The fishermen reported very good fishing today. They caught many fine
fish.
```

the words "fishermen" and "fishing" would be skipped, but the word "fish" would be found and highlighted.

Forward

Default: ON

When **on**, searches forward from the cursor, towards the end of the file. If **Entire Scope** is selected, the search starts at the beginning of the file.

Backward

Default: OFF

When **on**, searches backwards from the cursor, towards the beginning of the file. If **Entire Scope** is selected, the search starts at the beginning of the file.

Global

Default: ON

If **on,** searches the entire document for the find text.

Selected

Default: OFF

If **on**, searches only within marked text.

From cursor

Default: ON

If **on,** searches forward or backward from the cursor location.

Entire scope

Default: OFF

If **on,** searches the entire document from the beginning or end of the file.

Find Next

Continues the search operation, using the data already in the **Text to find** window.

Goto Entry

Where: **Phonebook/Search/Goto Entry**

See Also: Phonebook

Moves the selection to the phonebook entry specified in the **entry number** field in this dialog box.

Goto Line

See Also: **Search**

Default: Current Line

Moves the cursor to the beginning of the line specified in the **line number** field in this dialog box. The current cursor position is shown in the status line at the bottom of the File Viewer window.

Kermit Options

Where: **QmodemPro/Options/Protocols/Kermit**

A detailed discussion of the options is beyond the scope of this document. The default options provided in this dialog box are correct for most circumstances, and should *not* be changed except on the advice of the System Administrator of the host system you plan to call. If you change the Kermit option and need to restore them to the defaults used by *QmodemPro for Windows*, select the Default button.

Key Mapping

Where: **Options/Emulation Options**/Keyboard

See Also: Macro keys
Desktop Options
Revise Phonebook Entry
Programmable Keys

This dialog box appears when you select a key for reassignment. Programmable keys can be used to send macro commands to *QmodemPro for Windows,* to run a script, or to send commands to the modem, or to send text to the remote host.

For instance, you can program a function key to send your UserID and Password to a BBS from the system's Phonebook entry, or change port and parity settings on the fly.

Key Assignments

To select a key for reassignment, highlight the desired key with the tab key or the mouse pointer, and click the left mouse button. A dialog box will pop up, allowing you to enter the text, macro or meta command for up to four shift levels: **Normal, Shift, Ctrl** and **Ctrl-Shift**.

Edit Format

This set of option buttons allows you to enter your selection in character mode, decimal or hexadecimal.

Keyboard Mapping Options

Where: <u>Emulation Keyboard Mapping</u>

See Also: <u>Emulation Options</u>
<u>Macro Keys</u>

F10

By default, the **[F10]** key activates the menu bar. If you prefer to re-assign this key, change the default from **Mapped** to **Menu**.

Numlock

By default, the **[NumLock]** key toggles the numeric keypad. If you prefer to re-assign this key, change the default from **Keypad Toggle** to **Mapped**.

Key Selection

This option controls the behavior of the Keyboard window. If **Tab** is selected, you can select keys by pressing the **[Tab]** key to move from one keyboard button to the next. If **Keystroke** is selected, press the actual key on the keyboard you would like to edit.

Keyboard Type

Change this option to display the keyboard layout most closely matching your own. Two options are available: **101-Key** and **84-Key**.

Macro Keys

Where: **QmodemPro/Options/Macro Keys**

See Also: <u>**Emulation Keyboard Mapping**</u>
              <u>**Revise Phonebook Entry**</u>
              <u>**Desktop Options**</u>
              <u>**Programmable Keys**</u>
              <u>**Macros**</u>
              <u>**Macro System Functions**</u>

Keys 1 - 0

These buttons correspond with the number keys along the top row of your keyboard. These keys in the various alternate positions can store macros and commands. They also appear at the bottom of the *QmodemPro* window for easy selection. The leftmost button on the Macro Bar indicates the shift state: A, AS, AC, ASC for **[Alt]**, **[Alt][Shift]**, **[Alt][Ctrl]**, or **[Alt][Shift][Ctrl]**.

Assignments

To edit a macro key, highlight the desired key with the mouse pointer, and click the left mouse button. In the **Assignments** window, enter the text, macro or meta command for up to four shift levels: **[Alt]**, **[Alt][Shift]**, **[Alt][Ctrl]** and **[Alt][Shift][Ctrl]**.

Labels

Enter the text you would like to use as a label for the macro key you are editing. These labels are displayed in the buttons on the Macro bar at the bottom of the *QmodemPro* window.

Labels must fit on the Macro Bar keys and are limited in length, depending on the characters you use and the size of your terminal screen. Up to nine characters may be entered, but the number displayed may be limited. Check your labels to be certain they fit on your screen.

Edit Format

This set of option buttons allows you to enter your selection in character mode, decimal or hexadecimal.

Default

Macro key assignments can be saved in a macro file. This window shows the name of the current default macro file which is loaded when *QmodemPro for Windows* is started.

New

Clears all macro key definitions from the current dialog box so that you can edit and create a new macro file.

Load

Pops up a common dialog box allowing you to select a macro file to edit or use.

Save

Saves the macro key definitions in the dialog box to a macro file.

Set Default

This button sets the currently open macro file as the default. This is the set of Macro Keys in effect when you first start *QmodemPro for Windows.*

Close

Closes the Macro Keys dialog box. If you do not save your changes, the current macro key settings will only remain in effect until a new file is loaded, or until you exit *QmodemPro for Windows.*

Manual Dial

Where: **QmodemPro/Phonebook/Manual Dial**
Phonebook/Dial/Manual Dial

See Also: Phonebook

This dialog box allows you to enter a phone number for a manual dial. The data entered in the Manual Dial dialog box will be sent directly to the modem as if it were an actual Phonebook entry. The number is not checked for validity.

Number to Dial

Enter the **telephone number** you want to dial in this window. To enter special control characters, refer to **Special Control Characters.**

Device to Use

Select the **device** you would like to use from this list. The current device is the default. See the section on **Devices** for more information.

Dial

Click the **Dial** button to begin dialing. The **Manual Dialer** dialog box will pop up, showing you the progress of the current call.

Manual Dialer

The **Manual Dialer** dialog box displays the current status of the call being dialed.

Details

The **Details** window shows information about the entry currently being dialed. This information is extracted from the Manual Dial dialog box.

Progress

The **Progress** window shows the number of seconds to wait for a connection, the seconds between dialing attempts, the current time, and the start time for the current dialing attempt.

Status Messages

The **Status Messages** window shows a list of commands sent to the modem, responses from the modem, and connect messages. When a successful connection is established, the Dialer dialog box is replaced by the Terminal Screen, or the Fax Status dialog box, depending on the type of connection made.

Cycle

The **Cycle** button aborts the current dialing attempt and redials after pausing for the number of seconds shown in the **Seconds between** field in the Progress window.

Extend

Use the **Extend** button to extend the length of time *QmodemPro for Windows* will wait for a successful connection with the current entry. Use this if the **Wait for connect** counter is nearing zero and the system has not answered the phone or the modems have not yet established a connection.

Note that modems usually have an internal timer that may override the time extension. Refer to your modem manual for instructions on changing this internal setting.

Cancel

The **Cancel** button aborts the current dialing attempt and closes the Manual Dialer dialog box.

Manual Send Fax

> **Where:** **QmodemPro/Fax/Send**
>
> **See Also:** **Fax Send**
> **Sending and Receiving Fax Messages**

This dialog box prompts you for the information *QmodemPro for Windows* needs to send a Fax message manually.

Number to Dial

Enter the **telephone number** you want to dial in this window. To enter special control characters, refer to **Special Control Characters**.

Device to Use

Select the **device** you would like to use from this list. For more general information on devices, see **Devices**.

Fax Information

The following information is used to fill in Fax Variables in the cover sheet.

To

The name of the person to whom the Fax is addressed. This value is stored in the @TO@ Fax Variable, used in custom cover sheets.

From

Your name. This value is stored in the @FROM@ Fax Variable, used in custom cover sheets.

Subject

The subject of the Fax message. This information is stored in the @SUBJECT@ Fax Variable, for use in custom cover sheets.

Document ...

The name and path of the document to send. Use the **Edit** button to browse other filenames, paths and drives.

Font

Pops up the **Select Font** dialog box, allowing you to select the type style and size to use to send the text in your Fax documents.

Options

Use Fine Mode

> **Default: OFF**

Two resolution modes are supported, Normal (200 x 100 dpi) and Fine (200 x 200 dpi). Faxes sent in **fine mode** look better but take longer to send.

Include Cover Page

> **Default: ON** if cover page exists (see **Options/File Definitions**); **OFF** if no cover page found.

Turn this option **on** to send the cover page defined in **Options/Fax** before the selected document.

Send

Pops up the **Manual Dialer** dialog box and begins dialing the phone number entered in the **Number to Dial** field.

Path Definitions

Where: **QmodemPro/Options/Files/Path Definitions**

See Also: **Download**
Upload

Download

Default: C:\QMWIN\DOWNLOAD

This is the default path where *QmodemPro for Windows* will place files that you download from a host system. When you request a download, you have the option to accept or change this path on a temporary basis before the transfer begins.

Upload

Default: C:\QMWIN\UPLOAD

This is the default path which *QmodemPro for Windows* checks first for files when you request an upload. When you request an upload, you have the option to accept or change this path on a temporary basis before the transfer begins.

Script

Default: C:\QMWIN\SCRIPTS

This is the default path where *QmodemPro for Windows* checks for script files to compile or execute. When you request a script operation, you have the option to accept or change this path on a temporary basis before the operation begins.

RIP Icons

Default: C:\QMWIN\RIPICONS

This is the default path where *QmodemPro for Windows* checks for RIP icons. Individual phonebook entries can also have their own RIP icon paths.

Phonebook Options

Where: **QmodemPro/Options/Phonebook**

See Also: **Phonebook Window**
The Phonebook

The **Phonebook Options** dialog box allows you to specify the default name and path of the phonebook *QmodemPro for Windows* should use, and whether or not to open the Phonebook Window automatically when the program is started.

Default phonebook

Default: C:\QMWIN\QMWIN.PHN

Enter the name of the default phonebook file here, or use the **Edit** button to browse file names, paths and drives.

Load at startup

Default: OFF

If **on**, opens the Phonebook window automatically when *QmodemPro for Windows* is started.

Print Phonebook

Where: **Phonebook/File/Print**

See Also: The Phonebook

The Print menu allows you to format and print selected information from the current Phonebook. All Phonebook entries are printed. The Print Phonebook dialog box lists the available fields in a window on the left, and the selected fields in a window on the right.

To select a field to print, highlight the desired information with the mouse pointer, and click the left mouse button.

To add the field to the list, use the **Add** button or double-click on an entry. Use the **Up** and **Down** buttons on a selected field to change its position in the **Print fields** list. Use the **Remove** button or double-click on an entry to delete a field from the list.

When you are satisfied with your selections, use the **Print** button to begin printing.

Protocol Options

This section allows you to customize some options and toggles **on** the internal protocols.

Some file transfer protocols offer configuration options in addition to the general options discussed below. Each protocol button opens a dialog box allowing you to customize your settings.

The default settings established for each protocol are the recommended ones for most circumstances, and unless there is a clear reason to make changes, these settings should be left alone.

File Options

Overwrite old

Default: OFF

Auto increment

Default: ON

The **Overwrite old** and **Auto increment** toggles work in tandem to control how downloaded files are treated. When you are prompted for a filename in one of the download protocols, a check is made to see if the file already exists. If it does, the Auto-Increment flag is checked. If it is set **on**, then the filename is incremented, and the transfer begins. Incrementing a filename consists of changing the last character of the filename to a 0 (zero), then a 1, and so on.

If the Auto-Increment flag is set **off**, then the Overwrite Files flag is checked. If this is set **on**, the existing file is deleted and the transfer begins. If it is set **off**, then you are prompted for a different filename before the transfer can begin.

Zmodem -- The exception to the rule

When downloading using any protocol except Zmodem, the Auto-Increment flag is checked first, and then the Overwrite flag. With Zmodem, there is a feature called Crash Recovery that causes filename handling to be done a bit differently. It overrides the above mentioned flags under certain specific conditions.

If the file you are downloading already exists and the file on disk is smaller than the one being downloaded, there is a possibility that the file on disk is a partial download that has not been completed for some reason. In this case, Zmodem attempts to use Crash Recovery to pick up where you left off. Crash Recovery is only begun if the file being received is identical to the one on disk.

If the two files are not the same, then the original rules apply -- a check of the Auto-Increment flag, and then the Overwrite, if Auto-Increment is **off**. If Overwrite is also **off**, the file is SKIPPED by the Zmodem download.

Save partials

Default: ON

A Partial is created when the Save Partials flag is set **on** and a download aborts. The portion that has been received successfully is not deleted. If it is set **off**, then any partially downloaded file is erased when the transfer aborts.

View GIFs

Default: ON

Turning this option **on** opens the GIF file viewer during the download of a .GIF (CompuServe Graphic Image Format) file. This allows you to see the picture file as you download it. Depending on the speed of your computer and your video hardware, GIF file viewing during a download may slow down the file transfer slightly.

Batch Download Options

System Date
File Date

Default: File Date

These two options allow you to specify your preference of file date and time on batch transfers. If you prefer to have all downloaded files dated the actual day and time you received them on your computer, select **System Date**. If you prefer to keep the original date on the file, select **File Date**.

Protocol Specific Options

Some protocols have additional configuration selections. Each of the following buttons pops up a dialog box with configuration options appropriate to the protocol.

Zmodem ...

Pops up the **Zmodem Options** dialog box.

Kermit ...

Pops up the **Kermit Options** dialog box.

ASCII ...

Pops up the **ASCII Options** dialog box.

Replace Text

The **Replace** dialog box allows you to define the text to search for, the text you want to replace it with, and search options.

Find what:

Enter the text you want to find in this window.

Replace with

Enter the text you want to replace the found text with in this window.

Search options

Several options allow you to customize your search.

Case sensitive

Default: OFF

When toggled **on**, finds only text which matches the capitalization of the search text exactly.

Whole words only

Default: OFF

Searches for occurrences of the **Find** text only as whole words, in other words the found text must be surrounded by white space or punctuation.

Global

Default: ON

Searches the entire document for the **Find** text.

Selected text

Default: OFF

Searches for the **Find** text within a marked block of text.

Forward

Default: ON

Searches forward in the document, towards the end of the file.

Backward

Default: OFF

Searches backwards in the document, towards the beginning of the file.

From cursor

Default: ON

Begins forward or backward search from the cursor location.

Entire scope

Default: OFF

Begins forward or backward search from the beginning of the file or the end of the file, respectively.

Find next

Continues the search operation, using the data already in the **Text to find** window.

Replace

Replaces the current occurrence of the found text with the text in the **Replace with** window.

Replace all

Replaces all occurrences of the found text with the text in the **Replace with** window.

Revise Phonebook Entry

System

This is the **name** of the host system BBS, or on-line service.

Primary

This is the **phone number** to dial for this system.

Alt. #1 - 4

Some multi-user systems offer alternate telephone numbers. Enter up to four alternate numbers in these four fields. When *QmodemPro for Windows* dials an entry, the primary number is tried first. If it is busy or does not answer, the numbers in the **Alternate Number** fields are dialed in order. See   **Special Control Characters** for more information on how to enter control code sequences in macros, modem command strings, and scripts.

User ID

This is the **name** you use to log into this service. This might be RICK HEMING for a BBS, or 75236,3312 for CompuServe, or rheming for a mainframe system. This item can also be used in scripts by using the USERID function and in macros by using the function @USERID.

Password

This is your **password** for this service. Be careful to choose a password that others will not easily be able to guess. This item can also be used in scripts by using the PASSWORD function, and in macros by using the function @PASSWORD.

Note

This is the name and path of the **Note** file associated with this phonebook entry. The Note file can be any DOS file. Text files can contain notes about the current entry, and *QmodemPro for Windows* will search this file during a text search. Use the **Edit** button to select from different file names, paths and drives.

Macros

This is the name and path of the **Macro** file associated with this entry. Use the **Edit** button to select from different file names, paths and drives. When the entry is dialed, the macros from this file will be displayed on the macro bar at the bottom of the terminal window.

Script

This is the name and path of a **Script** file associated with this entry. A script file in this field will be executed automatically when *QmodemPro for Windows* connects with the system in this entry, unless the dial command was started from a script in the first place.

RIP Icons

This is the path for **RIP Icons**, which are used in RIPscrip emulation mode to display certain menu items. RIP Icons are installed automatically in the default RIP Icons path for the entire system, which is defined in **Options/Path Definitions**. The path entered here is checked first for the specific dialing entry, and then the system path is checked. If this path is left blank, only the system path is used.

Device to Use

This is the communication **device** associated with this phonebook entry. This is the device *QmodemPro for Windows* will use to dial this entry. You can set this to one of the special predefined device names "**-Default-**" or "**-Current-**" if you wish.

Emulation

This is the **Terminal Emulation** to use with this phonebook entry.

Transfer Protocol

This is the default **Transfer Protocol** to be used when uploading and downloading files to this entry. Note that you always have the opportunity to change your protocol selection when you upload or download a file.

Retries

This is the number of times to attempt to dial and connect to this entry. If the phone is busy, *QmodemPro for Windows* will keep redialing until it connects, or until the number of **retries** matches the number in this field. A 0 (Zero) in this field indicates infinite retries.

Fax device only

Check this box if this phonebook entry is to be used to dial a Fax machine rather than another modem. If this box is checked, the **Fax Send** dialog box will pop up when you begin dialing, to request additional information.

Change Icon

You can associate regular Windows icons with Phonebook entries, and select Icon and Icon List viewing modes for your phonebook. The default icon is the *QmodemPro for Windows* "Q". Additional Phonebook icons are in a file called BBSICONS.DLL. Use the **Edit** button to select from different file names, paths and drives.

Clear

This button resets the statistics in the **Statistics** window back to zero.

RIPscrip Options

This dialog box defines options specific to **RIPscrip** emulation mode. This is a graphical emulation offered by many Bulletin Board Systems, which supports the mouse for menu selections, and displays 16 color graphics in 640 x 350 resolution.

Font Type

Use TrueType fonts

Default: OFF

TrueType fonts have an advantage over the standard RIPscrip fonts in that TrueType fonts are higher resolution and look better on your screen. Plus, they are scalable, meaning they can be enlarged or reduced on the fly to the exact size required by your application.

Turn this option **on** to use TrueType fonts Times and Arial in place of the RIPscrip Serif and Sans Serif fonts when you're in RIPscrip emulation mode. But note the cautions in the following paragraphs.

Caution: The RIPScrip emulation is not designed to be used with TrueType fonts. If you elect to use the TrueType fonts as replacements for the standard RIPscrip fonts you may experience screen display problems. These problems come about since the size and shape of the TrueType substitute fonts do not match the RIP fonts pixel for pixel.

The types of problems you may experience includes slightly offset text placement and overlap of text when multiple fonts are used. In cases where the screen uses an open-face font that has been filled with a color, you will probably find that the entire screen fills with the color. This is because the point selected for filling is no longer located within the boundaries of the original font character.

If you find these problems are happening regularly on the systems you call you should make use of the real RIPscrip font characters rather than the TrueType substitutes. The RIP emulation is not designed to use TrueType fonts, and our option to do so is an effort to offer an alternative higher resolution.

Flood Fill Type

Default: Type 1

There are three levels of support for the RIP Flood Fill operation in *QmodemPro*. A flood fill is the process of filling an area with a color. The reason these different levels are necessary is because many video drivers available for Windows 3.1 have bugs in the flood fill support routines. The bugs will manifest themselves as corrupted RIP screens, General Protection faults during RIP screens, or simply a lockup during a RIP screen. The three levels of support are explained below.

Level 1

This level uses the flood fill operation provided by Windows. This assumes that there are no bugs in Windows or the video driver. If there are bugs in the video driver they will show

up when using this level. This option must be used under OS/2 or Windows NT or any operating system other than Windows 3.1.

Level 2

This level bypasses the Windows flood fill services and directly calls the video driver flood fill support routines. If there are bugs in the video driver, they will probably show up at this level, as well as at Level 1.

Level 3

This level bypasses both Windows and the video driver flood fill support routines and retrieves the pixels directly from the video driver. This level is guaranteed to work under Windows 3.1.

How to choose the best level

If you are experiencing problems with certain RIP screens (incorrect displays, GP faults, or lockups), and you are using Windows 3.1, the best level is probably level 3.

If you are using OS/2 or Windows NT or any other operating system other than Windows 3.1, you must choose level 1 (the other levels will actually be unavailable to you).

Select Device

Devices

This is a list of all the devices you have defined. Select the device you want to use by moving the mouse pointer to the desired device, then click the left mouse button.

Select Download Protocol

Where: **QmodemPro/File/Download**

See Also: **Download**
Protocol Options
Uploading and Downloading Files

Protocol

The **Protocol** list shows the available file transfer protocols. Be sure to select the same protocol on both the host and local systems. See **Transfer Protocols** for a detailed discussion of protocols and configuration.

Select Emulation

Where: **QmodemPro/Terminal/Emulation**

See Also: **Emulation options**

Emulations

Select the emulation you want to use by moving the mouse pointer to the desired emulation, then click the left mouse button. See **Terminal Emulations** for more detailed information on this topic.

Select Font

This dialog box allows you to change the default font (typeface), style and point size. This dialog box occurs in several places. In the Text Editor and File Viewer, the **Font** selection changes the appearance of the text in the Viewer or Editor window.

In the Fax Send and Manual Fax Send dialog boxes, the **Font** selection defines a particular typeface and style to use when sending a text file as a Fax message.

The window at the bottom of the dialog box shows a sample of your selected type font.

Font

Default: Terminal

This is the name of the Windows type **font**, for instance Courier or Terminal. The selection of fonts depends on what fonts you have installed from the Windows Control Panel font manager. Monospaced TrueType fonts such as Courier New are supported, and are denoted by a "T" to the left of the font name.

Font Style

Default: Normal

The **Font Style** can be normal, *italic*, **bold** or ***bold italic,*** depending on the styles available for a particular font. Italic, bold and bold italic are used for emphasis, while normal is used for regular text.

Size

Default: 12

The **Size** is measured in points, approximately 1/72 of an inch. The text you are reading right now, for instance, is 10 points, measured from the top of the highest letter, for instance the letter "f" to the bottom of the lowest letter, for instance, the letter "g".

12 point type is roughly equivalent to Pica or 10 pitch on a typewriter.

Sample

The **Sample** window displays the type font, style and size as it will appear in the viewer or editor window, or in your Fax document.

Select Icon

This dialog box allows you to view the icons in an .ICO, .DLL or .EXE file, and select the icon you would like to associate with a Phonebook entry.

To look at icons, select a file name from the **File Name** window, or browse other directories and drives for more selections. Your icons will appear in the window under the **Total Icons** legend. Double-click the icon you would like to select for your phonebook entry.

Sort Phonebook

You can sort phonebook entries based on the information in any combination of fields. For instance, to sort by area code, enter **Primary Number** as the first sort field. To sort by system name, select **Service** as the first sort field, and so on.

Phonebook fields

This list shows the names of all the fields in a phonebook entry. Select an entry by clicking on it with the left mouse button. Double-click to move the selected entry to the Sort Fields list, or use the **Add** button.

Sort fields

This is the list of fields on which the Phonebook will be sorted. If multiple fields are placed here, second level sorting or deeper is performed. For example, sorting by Phone and System results in a list by number, and within each area code, by System Name. Double-click on a field to remove it from the sort criteria.

Sort Method

Default: Ascending

These two options determine in what order the sorted entries will be arranged.

Ascending

Ascending puts letter A at the top and letter Z at the bottom, like a dictionary. Numeric entries are sorted with 0 at the top and 9 at the bottom

Descending

Descending sorts backwards, from Z to A and from 9 to 0.

Add

Use the **Add** button to move a phonebook field to the Sort Fields list.

Remove

Use the **Remove** button to delete a phonebook field from the Sort Fields list.

Up

Use the **Up** button to move the selected field in the Sort Fields list towards the top of the list.

Down

Use the **Down** button to move the selected field in the Sort Fields list towards the bottom of the list.

Sort

Begins the sorting operation. The sorted phonebook will replace the original version of the Phonebook file.

Sounds

Where: **QmodemPro/Options/Sounds**

See Also: **Customizing the Desktop**

If your PC is equipped with a sound card or a sound driver for the PC speaker, and the appropriate drivers are installed in Windows, *QmodemPro for Windows* will play .WAV files to notify you of certain system events.

Sounds

This window lists all the sounds, with their associated WAV file name. Select a sound to edit by clicking on it with the left mouse button.

Change

You can assign a sound to each event by selecting the event you wish to modify, then clicking the **Change** button. This pops up a file selection dialog box, allowing you to select and play a .WAV file.

Test

The **Test** button plays the selected sound file.

Tab Size

Where: **File Viewer/Options/Tab Size**
Text Editor/Options/Tab Size

See Also: **The File Viewer**
The Text Editor

Tab size 2 - 20

Default: 8

Sets the number of spaces assigned to each tab character when Tab expansion is toggled on. Enter the number of spaces in the text window.

Upload File Selection

Where: <u>**QmodemPro/File/Upload**</u>

See Also: <u>**Upload**</u>
             <u>**Upload Status**</u>
             <u>**Download**</u>
             <u>**Protocol**</u>
             <u>**Uploading and Downloading Files**</u>

This dialog box allows you to select a transfer protocol, and select single or multiple files to upload to a host system, BBS or on-line service. The default upload drive and directory are taken from the **Upload Path** in the <u>**Path Definitions**</u> dialog box.

Send File(s)

Sends (Uploads) the file or list of files in the **Send Files** window. Wording and selection options depend on whether a single file protocol such as Xmodem is being used, or a batch file protocol such as Ymodem or Zmodem.

Protocol

The **Protocol** list shows all the available file transfer protocols. Be sure to select the same protocol on both the host and local systems. The appearance of this dialog box will change according to whether a batch or non batch protocol is selected. Non batch protocols such as Xmodem do not allow you to select multiple files to send.

Send List (batch protocols only)

This is the list of files that will be sent (uploaded) when you select the **Send List** button. See the **Add** command below to add files to this list.

Add

Adds the files selected in the **File Name** window to the **Send Files** window.

Add List (batch protocols only)

Adds to the **Send Files** window a list of files saved using the **Save list** command.

This is a standard Windows <u>Common Dialog Box</u>.

Remove (batch protocols only)

Removes a file from the list of files in the Send Files window.

Save List (batch protocols only)

Saves the list of files in the **Send files** window to a file on disk. See the **Add list** command above.

Upload Status

Where: **QmodemPro/File/Upload/Upload File Selection/Send List**

The upload status window shows the state of the current file transfer. See **Download Status** for an explanation of each item in this dialog box.

Wrap Column

Where: **Text Editor/Options/Wrap at column**

See Also: **The Text Editor**

Sets the right margin for automatic **wordwrap**. If you are using the editor to compose text to upload into a message on a BBS or on-line service, the recommended setting is column 72.

Zmodem Options

Where: **QmodemPro/Options/Protocols/Zmodem**

See Also: **Upload**
Download
Protocol Options
Uploading and Downloading Files

CRC 16
CRC 32

CRC (Cyclical Redundancy Check) is a very reliable method of error checking. Two methods of CRC calculation can be used for Zmodem transfers, 16 bit and 32 bit. In most cases 32 bit CRC is used. In fact, the CRC type is automatically negotiated by *QmodemPro for Windows* and will switch to 16 bit when 32 bit CRC is not supported by the remote system. This option allows forcing 16 bit if special circumstances require it.

Autostart

When Auto-Start is enabled, Zmodem file downloads are started automatically whenever a Zmodem download transfer header is received. If this parameter is disabled, Zmodem downloads must be started manually with **File/Download** command.

Registration Information

This information will be shown in the *About QmodemPro* dialog box, on the Help menu.

Name

Enter your name, as you would like it to appear in the Registered To portion of the window.

Company

Enter your company, as you would like it to appear in the Registered To portion of the window. You may leave this blank if you prefer.

Registration Number

Enter your Registration Number here, in the format XX-XXXX. This number is printed on the registration card bound into your manual. Please be sure to fill in and mail the registration card as soon as possible.

You will need to provide this registration number to MSI staff should you require technical support.

ANSI Options

Enable Ansi Music

This controls whether or not ANSI music will be played through the PC speaker when it is received. *QmodemPro for Windows* will play music if this option is toggled **ON**, and encoded ANSI music is received while on line.

To be recognized by *QmodemPro for Windows*, ANSI music must conform to the following format:

```
ESC [M . . . . . music codes . . . . ^N
```

Please refer to your BASIC language manual for further information on the PLAY statement. Music strings must end with the ^N (Ctrl N) character. If the ^N character is missing, it is assumed that the following text, up to 255 characters, is playable music.

Troubleshooting

Sometimes *QmodemPro for Windows* may not work for you the way you believe it should. In this section, we present a troubleshooting guide to some of the more common problems, and their solutions.

Before calling technical support with a problem, we recommend you spend some time reviewing the following list -- it could save you some time and money, and get your system back to top performance.

Modem problems

It says "Carrier detected" or "On-line" when I'm not connected

Modem doesn't hang up

Modem doesn't dial

Modem disconnects in the middle of a call

Dropped characters

File transfer errors

File transfers slow

File transfers abort

Garbled screens

Other hardware problems

Changing your port settings in Windows

Software problems

Error messages

General Protection faults

RIPscrip problems

Lockups

Fax Problems

Technical Support

Technical Support Services

Mustang Software, Inc. strongly believes in providing quality software at a reasonable price. We back our products with technical help when our registered users need it. Please send in your registration card to enroll in our technical support plan.

Until we receive your registration information, our staff cannot provide you with technical support via our BBS or by phone.

Benefits of Sending Your Registration

- Free technical support by phone Monday through Friday between 9 a.m. and 5 p.m. Pacific time
- Automatic notification of major program updates
- Additional program updates at a reduced rate
- Periodic mailings of the *MSI Monitor*, Mustang Software's newsletter. The *Monitor* contains information about new product releases and updates, technical tips, and other product information.
- 24-hour access to the private multi-line Registered User's *Wildcat!* HQ BBS at Mustang Software
- BBS access to download utility programs written by MSI staff and other authors which enhance and add features

Technical Support

QmodemPro for Windows is designed to be easy to use, and this manual should contain the answers to most of your questions. Read it first. Make use of the on-line help, and check the appendix for troubleshooting procedures. If you're still stumped, technical support is only a phone call away.

You may first want to call our private *Wildcat!* BBS for Registered Users at 805-873-2400, and leave a quick question for our technical staff. This is a particularly useful way to receive expert guidance with the more advanced features.

Note: If you have your registration number handy, you can get full access on your first call. The System Operator (Sysop) will make your access permanent if you leave a message requesting upgraded access.

Another alternative is CompuServe where we sponsor a part of the PC Vendor Support Forum. You reach us by typing GO PCVENA at any CIS prompt and then selecting Subtopic 9. Our PIN is 75236,3312.

Support is also available on GEnie in the Qmodem Support Roundtable. Type QMODEM at any GEnie menu prompt to reach the support section. We can also be reached at America On-line, in the Mustang forum.

Support is not available by mail or Fax.

If you are still unable to find the answer to a question--or if you need a quick explanation, please call us Monday to Friday between 9 a.m. and 5 p.m. Pacific time.

When calling for support please:

Have your registration number handy (it is printed inside the front cover of your manual).

Be at your computer with your manual handy.

 Be ready to describe the problem in detail.

 If possible, be ready to duplicate the problem on your system.

 You can call technical support at (805) 873-2550.

QmodemPro for Windows is constantly undergoing enhancements and revisions. This is normal software maintenance. The current version of *QmodemPro for Windows* may be a major or minor release. We encourage you to keep your software updated--we can provide the best support to users who are running the current major release. The cost is minimal and the benefits are great.

Modem problems

Most problems with a communications program originate with the modem. It pays, therefore, to begin troubleshooting a problem by verifying that the modem itself is operational and connected properly to the PC.

Checking external connections

If you believe the modem is not working properly, check all external connections first. If you use an external modem, be sure that the modem is turned **on** and its power supply is plugged into an electrical outlet. Verify that the serial cable connecting the modem to the PC is securely plugged into both devices. And be sure the telephone line is plugged into the wall outlet and the modem.

If you use an internal modem, check the phone line connection, and verify that the modem is securely seated in its slot inside your PC. Always turn **off** the power before opening your PC!

Modem setup problems

If your modem still doesn't work the way it should after you've verified that it is connected properly, the next step is to investigate and correct the modem's internal settings.

First, let's verify that *QmodemPro for Windows* is communicating correctly with the modem.

Start *QmodemPro for Windows*, wait for the terminal screen to appear, and type

    AT **[Enter]**

at the terminal screen. If the modem is connected properly, it should answer

    OK

or

    0 (zero)

If it doesn't respond, here are some possible reasons:

   The modem is attached to a different COM port than *QmodemPro for Windows* is expecting. Try changing the software to look at the correct port.

   The modem is using the same IRQ (Interrupt ReQuest line) as another device (most commonly the mouse), and the other device is taking over the interrupt and preventing communication with the modem. Consider moving either the modem or the other device to a different port and IRQ to avoid further conflicts.

QmodemPro for Windows says "Carrier detected" or "On-line" even when I'm not connected

The most common problem here is Carrier Detect being forced **on** by the modem. If the modem in question has a DIP switch for this function, set it so CD (Carrier Detect) is NOT forced **on**. Individual modems say this differently, and may range from "Follow line signal" to "Reflect Reality". Change the setting to have Carrier NOT forced, to Follow Line Signal, or to Reflect Reality. All of these indicate that is will NOT be forced into an "**on**" state all the time, but will only be present when a caller is actually on-line.

If your modem DOES NOT have a DIP switch for this function, try sending the command

> `AT&C1` **[Enter]**

If the problem persists even after checking these two items, there is but one other possible problem, and it usually isn't easy to troubleshoot. If CD is still seen it suggests that there is a second RS-232 port (or other device) making use of the same address that you are using. This is most commonly the result of inserting an internal modem addressed as COM2 in a PC that already has a COM1 and a COM2 either on the system board or on an add-in card.

The solution is to DISABLE the other COM2 so there is no conflict or confusion of the signals. This is usually done by either changing dip switches on the system board or add-in card, or removing a jumper over two or more pins. You will need the manual for the PC or the add-in card to perform this operation, and our technical support staff can help if you have the manuals and the hardware when you call.

It is important to realize that the conflicting or duplicate COM port does not even have to be operational, there only has to be an optional provision for addition of the offending port for it to be a conflict if not disabled. For example, many Multi-I/O boards come with a serial port and a game port, and so on, and offer an additional serial port which doesn't even have the chips installed unless you pay extra. However, the board often activates the jumpers for the second port, making it available at the address as a duplicate.

Modem doesn't dial

Does the modem have a speaker? Is the speaker turned on? Is the volume control turned up? The command to turn the modem speaker **on** during dialing is:

`ATM1` **[Enter]**

Raising and lowering the speaker volume is done in one of two ways. Some modems have a volume control knob, which you can adjust in the same way as your radio. Turn the volume high enough to hear the dial tone and connection negotiation. Other modems control speaker volume via a software command. This is often:

`ATL1` **[Enter]**

Other numeric values may be supported -- generally speaking the higher the number, the louder the volume. Consult your modem manual for exact instructions.

Now try dialing the modem again. If it is working correctly, it will pick up (you hear a dial tone) and you will then hear tone or pulse dialing, the remote phone ringing, and a high pitched tone and hissing noise as the modems make the connection.

If you do not hear tone or pulse dialing, verify that you have entered the proper Dial String in the terminal command setup for your modem. The correct command for tone dialing is

`ATDT`

The correct command for pulse dialing is

`ATDP`

If you hear tones, but the call does not go through, it is possible your phone line does not allow tone dialing. Change the dial string to pulse (ATDP) and see if the call goes through.

If it still doesn't work, try putting an ordinary phone on the line, pick up the handset and dial an outgoing call. If the phone works and the modem doesn't, verify that the phone cord is plugged into the correct outlet on the modem. There are usually two outlets, one labeled "WALL", and the other labeled "PHONE". Not all modems are labeled, however. Try swapping the cord to the other jack, and repeat the test.

If the phone doesn't work, there may be a problem with another extension phone on the same line, or a problem with the phone line itself. Unplug all the other phones on that line, and try it again. If that doesn't help, there is probably a problem with the phone line. Ask your phone company repair office for assistance.

Modem doesn't hang up

Many modems come from the factory with the DTR signal forced TRUE (always **on**). *QmodemPro for Windows* wants to control the DTR signal to hang up the phone, since this is the most reliable method.

Some modems have dipswitches to control DTR, while others hold this information in their Non-Volatile RAM (NRAM). Check your modem manual to be sure of the correct procedure for setting DTR to follow line conditions.

On modems with NRAM, the command is usually

`AT&D2` **[Enter]**

Then go into the **Device/Modem** dialog box, and change the **Hangup** command to "DTR".

Modem disconnects in the middle of a call

Does your phone line have Call Waiting or Call Alert service to inform you of incoming phone calls while you're talking to someone else? The click or beep sound notifying you of an incoming call is often enough to break a modem connection. Find out if your phone company offers "cancel call waiting" service -- this is usually accomplished by dialing a special code before placing an outgoing call, and is active for the current call only.

Did someone pick up an extension phone in your home, while you were on-line? This frequently causes data calls to disconnect abruptly.

Are your telephone lines exceptionally noisy? When you pick up the phone, do you hear noises like static, whistling, rustling, or howling? All these sounds can interfere with data communications, and many modems give up and disconnect when line noise becomes very severe. Check with your phone company's repair service to see if they can fix poor line quality.

Some long distance calls can be routed over noisy phone lines -- in this case the best thing to do is hang up immediately and try placing the call again. Chances are your next call will be routed differently, and line noise will be less of a problem.

Incorrect flow control settings can also cause the modem to disconnect when it gets more data than it can handle. If you have a high speed (9600 baud or faster) or error correcting (MNP) modem, verify that CTS/RTS flow control is turned **on**. Verify also that the port speed or DTE is LOCKED at the appropriate value -- generally 19200 or 38400 for most 9600 baud or faster modems).

Use the Quick Device Config for your modem to reset it to the correct operating values.

Dropped characters

Sometimes characters are dropped on the way from the modem to the computer. This problem is far more common with external modems than internal ones, which should already have the correct type of serial port chip installed on the modem itself.

The end result of character dropping is that letters are missing from words, graphic screens are messed up, with some lines shorter than they should be, and raw ANSI code appears on the screen instead of being interpreted as colors. During uploads and downloads, the same problem causes transfer errors and resent blocks.

This is because for one reason or another, the modem is too fast for the serial port. If you have a high speed modem (9600 baud or faster), but the serial port is not designed to handle speeds higher than about 2400 baud or so, characters will be lost from time to time.

The ultimate fix is to replace the serial port chip with a 16550 buffered UART, but in the interim you could try some of the following suggestions:

Add the following lines to your SYSTEM.INI file under the [386enh] section:

```
ComXBuffer=1024
ComBoostTime=30
```

Replace the letter X with the appropriate COM port number, 1-4. This increases the size of the buffer and helps prevent dropped characters. Use any standard ASCII editor to add this line, including the **File/Edit a File** menu choice. SYSTEM.INI is located in your Windows directory.

Next, from the Windows Control Panel, select "386 Enhanced". Change the "Minimum Timeslice" to 30 (the default value is 20). This may help resolve some timing-related problems by allowing each task more CPU time.

Your disk cache may be interfering with data from the COM port. Try disabling "staged writes" or "delayed writes" in your disk cache, and see if the problem improves.

Try lowering the DTE speed -- if it's locked at 57600 your computer may in fact not be able to keep up with the modem, and characters will get lost. You won't notice any significant change in throughput by locking at 19200 or 38400.

Consider upgrading the Windows communication drivers by purchasing a high performance com driver package such as Turbocom or Cherry Hill. Information about high performance com drivers is at the end of this section.

Garbled screens

There are several possible causes here. If the garbled screens begin as soon as you start *QmodemPro for Windows* it usually indicates that your mouse is connected to the serial port you have specified for your modem. You may have mistakenly indicated the wrong serial port, or your mouse may be using the same IRQ as your modem (especially if you have an internal modem and a serial mouse). The solution to this problem is to determine which IRQ is being used for your mouse and your modem, and to change one or the other to avoid the duplication.

The other most common cause is related to the problem discussed previously -- characters come into the serial port faster than the computer can process them, and they are lost rather than held in a buffer. This causes ANSI screens to break up and display the raw ANSI code (arrows, square brackets and numbers) rather than interpreted color attributes.

The next likely cause is line noise. Lower speed modems (300, 1200 and 2400 baud) are more susceptible, since most of these do not have built-in error correction.

Since the modem converts binary digits into sound, then transmits those sounds over phone lines, any stray noises added by the phone company's switches and circuits could be interpreted by the modem as data.

Line noises causes stray characters such as curly brackets and check marks to be interspersed with the text you expect to see on screen. If line noise is a continual problem, consider asking your local phone company's repair service to investigate, or upgrade to an error correcting modem which filters out these stray characters.

Software settings are another possible cause for garbled screens. There are a couple of settings in the *QmodemPro for Windows* configuration itself which might cause problems. The typical symptom here is that you see letters and numbers instead of lines and block graphics, or odd high-ASCII characters in words.

One possible cause is that you're logging on at 7 bits instead of 8 bits (since the letters and numbers you see are the graphic characters without the 8th bit). You can change this by changing the phonebook device entry to 8-N-1 from 7-E-1 for systems affected by this.

Another possibility is that you've got **8th bit strip** turned **on**. When you're connected to a system that is doing this, select **Emulations/8th Bit Strip** and see if the screens look better.

Finally, it could be that you are using TTY terminal emulation instead of ANSI, and *QmodemPro for Windows* is thus not interpreting the ANSI escape sequences (a lot of [brackets and numbers) as graphics and color. Try changing the emulation by selecting **Terminal/Emulations** and select ANSI.

File transfer errors

File transfer problems can be difficult to trace, since there is as much likelihood the problem is on the remote system as on your own system.

You should suspect an error in the way you have set up *QmodemPro for Windows* if you have the same problems on every remote host you call.

File transfer status window

The upload and download status windows display the current status of the transfer -- name and size of file (if known), number of bytes transferred, characters per second, and percent completed (if known). Protocol error messages are also displayed in this window, and may provide some clues about the problem.

Note that the term Download always refers to receiving a file onto your PC from a remote computer, and Upload refers to sending a file from your PC to a remote computer.

Some modems have separate flow control settings for sending and receiving. Check your modem manual for information, and be sure both sending and receiving flow control are turned on properly.

Download errors (receiving files)

If uploads work and downloads don't, this is an indication of a flow control or buffer overrun problem. Check your flow control settings in *QmodemPro for Windows*, and check the modem itself if it has flow control settings.

UART overrun occurs when the modem receives data faster than the computer can process it, and any bits not processed are lost instead of being stored. When bits are lost in this way, the UART signals the PC that an overrun error has occurred.

A buffered UART chip (the NS16550AFN) is a relatively inexpensive solution to the UART overrun problem. This is a regular 40-pin IC chip which replaces the 16450 or 8250 chip on your serial board, and is available from most electronic supply houses. Or try lowering the DTE lock rate.

Upload errors (sending files)

If downloads work and uploads don't, the problem most likely originates with the remote computer -- especially if the problem is not evident on more than one system you call. There is not much you can do at your end, other than leaving a message to the system operator indicating you're having upload problems. Always include as much detail as you can in your message -- the transfer protocol you used, the baud rate, any error messages you saw while uploading, and the approximate number of bytes transferred before the errors occurred.

If you experience upload problems on most of the systems you call, the problem is more likely to be with your own modem's configuration -- some modems have separate flow control settings for both receiving and sending. Consult your modem manual for help if you suspect this may be the case.

File transfers slow

Some <u>transfer protocols</u> are inherently more efficient than others, and protocol selection therefore has a great bearing on how fast *QmodemPro for Windows* transfers your data. Using an efficient protocol such as Ymodem or Zmodem and clean phone lines, you should expect throughput of around 1000 to 1100 characters per second (CPS) at 9600 baud, 1500 to 1600 CPS at 14400 baud. 2400 baud connections without error correction should average around 230 characters per second.

Just as with automobiles, however, your mileage may vary. File transfer speed can be influenced by many factors, from the speed of your computer to the number of active Windows applications you have opened. So long as the transfer window is not reporting errors, you should not be too concerned about transfer speed unless it is more than 10 to 15 percent slower than the figures quoted above.

Because transfer protocols perform error detection during a transfer, blocks of data which are not received correctly will be retransmitted until they are received correctly. Depending on the amount of retransmitted data, this will definitely slow down the overall speed of the transfer, but in the vast majority of cases the file is eventually transferred correctly.

Some conditions which commonly contribute to file transfer errors are phone line noise (static on the line interpreted as data by the modems), incorrect flow control settings, and UART overruns. The protocols are designed to terminate (abort) the transfer after a certain number of consecutive errors.

File transfers abort

File transfers may abort because too many errors were detected by the transfer protocols, as discussed above. There are other reasons a transfer may terminate unexpectedly, however.

Someone in your house might pick up an extension phone while you are on-line -- this almost always causes a file transfer to abort, and may break the connection altogether! Call Waiting service on your phone line will almost certainly cause a file transfer to stop if an incoming call comes during an on-line session. Or the system operator on the remote computer may abort the transfer manually.

Another possible cause might be a sudden burst of line noise, although this is more likely to be a problem with non-error correcting modems. Modem error correction helps a tremendous amount with noisy phone lines, but it is not infallible, and extreme line noise can interrupt file transfers whether or not your modem has error correction built in.

Some hardware related causes for aborted transfers include standard DOS errors such as "File not found" or "Disk full", and problems with your memory and drives. Check your available disk space before receiving a large file, and check the spelling of a filename and path before trying to upload.

Changing your port settings in Windows

Modems are not the only accessories that can be connected to a COM port. If your computer has a Mouse, Trackball or other Pointing Device, it may be connected to a COM port. Also, some kinds of printers are connected to Serial ports rather than Parallel ports.

When you install your modem, you may need to find out what other devices are connected to communication ports in your PC. Each device needs its own address, but most devices also allow you to select additional addresses by moving switches or jumpers on an expansion card, or by running the setup utility for your computer.

Because there are so many different devices available for PCs, and different ways of setting them up, it's not possible to go into all the variations here. If, after installing your modem, you feel it is not working correctly, the fastest way to solve the problem is to call our Technical Support lines. Our technicians are experienced and knowledgeable, and they want to help you use and enjoy *QmodemPro for Windows*.

To use your modem on COM3 or COM4, you may need to make some changes to the port settings in Windows in order for *QmodemPro for Windows* to work correctly. You may also need to change the IRQ settings on your serial card or internal modem, if your Windows communication drivers do not support more than one device per IRQ.

To do this, go to the Program Manager, Main Program Group. Double click the **Control Panel** icon, and select **Ports** from the window that pops up. From there, select the port you want to modify, go to **Advanced** settings, and make the appropriate changes to the Base Address and IRQ.

Refer to your Windows documentation, and the manual for your modem, serial card and communication drivers for information on how to set and select IRQ and Base Address.

Other hardware problems

In rare instances the modem itself, or the serial port or cable, may be defective. You should consider this a possibility only if the preceding troubleshooting suggestions do not work, or work very inconsistently.

Substitute a different modem if you can, and see if the problems persist. Replace the modem cable -- an intermittent bad connection inside the cable can cause problems every time the modem or cable is moved. Modem cable problems can be especially difficult to track down. Try a different serial port card, too.

If the modem will not work properly no matter what you do, your next step should be to contact the modem manufacturer's own technical support department. They will be able to provide up-to-date information on known problems with your modem model, and in many cases offer inexpensive or free ROM updates.

Software problems

In most cases, *QmodemPro for Windows* should run correctly as installed, without modification to its basic setup. Be sure you understand the implications before you make changes to the basic operational parameters of the program, and make note of the settings you change, so that they can be restored to their original state if necessary.

Error messages

General Protection faults

RIPscrip problems

Lockups

Error messages

When a program error occurs, an information window will pop up over your screen with a brief description of the error. The error message may be generated by *QmodemPro for Windows*, or by another Windows application, or by Windows itself.

There are two basic types of program errors. A fatal error means that the problem was severe enough that the application could not recover, and must be closed down. Fatal errors are not necessarily the fault of *QmodemPro for Windows* -- other applications can sometimes unexpectedly use memory or devices belonging to *QmodemPro for Windows* and cause a crash.

A serious error will probably require you to close all applications, exit Windows, and restart your computer, so you should save your work in other windows, if you can, before rebooting.

A non-fatal error is one which indicates that needed configuration information is not present, or conflicts with other settings, or that an error was encountered that was not severe enough to shut down the program. The error messages will indicate the nature of the problem, and may suggest how to fix it.

General Protection faults

A General Protection or GP fault occurs when an application overwrites memory belonging to another application. Since the memory has become corrupted, Windows steps in and terminates the application.

It is not always easy to tell which Windows application caused the GP fault. Sometimes the error message window will provide an application name and a memory address, but this information is not always reliable. The best thing to do when you get a GP fault is to save your work in your other applications, and exit and restart Windows as soon as possible.

If you get a lot of GP faults when you use *QmodemPro for Windows*, make note of the steps you took immediately before the program crashed, and report the details to MSI Technical Support.

RIPscrip problems

If you consistently experience GP faults in <u>RIPscrip mode</u>, the most likely cause is the Flood Fill mode you've selected. There are three levels of support for the RIP Flood Fill operation in *QmodemPro*. A flood fill is the process of filling an area with a color. The reason these different levels are necessary is because many video drivers available for Windows 3.1 have bugs in the flood fill support routines. The bugs will manifest themselves as corrupted RIP screens, General Protection faults during RIP screens, or simply a lockup during a RIP screen. The three levels of support are explained below.

Level 1

This level uses the flood fill operation provided by Windows. This assumes that there are no bugs in Windows or the video driver. If there are bugs in the video driver they will show up when using this level. This option must be used under OS/2 or Windows NT or any operating system other than Windows 3.1.

Level 2

This level bypasses the Windows flood fill services and directly calls the video driver flood fill support routines. If there are bugs in the video driver, they will probably show up at this level, as well as at Level 1.

Level 3

This level bypasses both Windows and the video driver flood fill support routines and retrieves the pixels directly from the video driver. This level is guaranteed to work under Windows 3.1.

How to choose the best level

If you are experiencing problems with certain RIP screens (incorrect displays, GP faults, or lockups), and you are using Windows 3.1, the best level is probably level 3.

If you are using OS/2 or Windows NT or any other operating system other than Windows 3.1, you must choose level 1 (the other levels will actually be unavailable to you).

Lockups

A computer is said to be locked up when it is completely unresponsive to the mouse or keyboard. Keep in mind that when a program is busy performing some operation, it may not respond for a period of time -- usually when this is the case you will see an hourglass icon on the screen rather than the usual mouse pointer.

If your computer seems to have locked up, wait a couple of minutes and try again. If it has not returned to life after that period of time, you will probably need to reboot and restart Windows. Rebooting the PC will cure the lockup, but any unsaved work will be lost!

It is unusual for a program such as *QmodemPro for Windows* to crash or cause your computer to lock up completely. It has been thoroughly tested under a variety of operating environments and conditions to ensure its reliability, but some unforeseen combinations of hardware and software can cause problems.

If you are able to make *QmodemPro for Windows* crash or lock up your computer repeatedly, please convey this information to MSI Technical Support, with as much detail as possible about your system configuration and the exact sequence of keystrokes and operations which led to the lockup.

Fax Problems

Sending and receiving Fax documents using a Fax modem can sometimes be troublesome. If you have difficulty try these suggestions.

Check the **Options/Device/Hardware/Baudrate** (not the Fax baudrate) and set it to 19200. This has proven to be the best selection for almost all Fax modems. The port may be locked at this rate. The Fax baud rate only controls the starting speed for Fax, which will be adjusted by the two Fax devices. 9600 is the standard Fax baud rate.

Try closing all other applications while sending documents and don't minimize *QmodemPro* when sending or receiving documents. Fax transmission is very timing dependent. When sending Fax documents you may experience problems if you have several other applications active at the same time, or if you try to send documents while *QmodemPro* is minimized and your system is performing other activity at the same time.

Note that Fax documents cannot be sent using Int 14, Digiboard or other drivers. Only Serial devices are supported. If the port (in **Options/Device/Hardware**) is set to anything but a Serial selection, you will be unable to set or change any FAX options.

SLIQ Script Language Interface for QmodemPro

This is an operational reference to the *QmodemPro for Windows* Script Language. All script language commands are listed in this help file, along with the information youre most likely to need while youre writing or editing a script. For more detailed instructions, including example scripts, please refer to Book 2 of the printed manuals.

Script Language Commands

Technical Overview of SLIQ

The SLIQ Debugger

Using Dialog Boxes

Using DLL Functions

Compiler Error Messages

Runtime Error Messages

SLIQ Script Language Commands

$

A

B

C

$INCLUDE Directive

Includes another source file into the current source file.

Syntax

```
'$INCLUDE 'filename'
```

Remarks

The $INCLUDE directive is an instruction to the script compiler to include the named script file into the current file. This is useful if you have a standard set of subroutines and functions that you want to share between multiple scripts.

Example

```
[in file a.scr]
'$include 'b.scr'
call test

[in file b.scr]
sub test
  print "Hello, world!"
end sub
```

+ (Concatenation) Operator

Concatenates (combines) two or more strings into one string.

Syntax

```
string1 + string2 [+ string3 ...]
```

Remarks

This operator is used to combine strings into a single, longer string. For instance, you can concatenate DATE and TIME variables into a single string variable called NOW (see example).

No spaces or other characters are placed between the strings during concatenation.

See also

LEFT, MID, RIGHT

Example

This example combines the current date and time strings into one string called now, and prints it on the screen.

```
dim now as string
now = date + " " + time
print now
```

ABS Function

Returns the absolute value of a numeric expression.

Syntax

```
ABS(number)
```

Remarks

number can be any valid numeric expression.

The absolute value of a number is the quantity of the number without regard for its sign (positive or negative). A negative number has the same absolute value as the corresponding positive number. In other words,

abs(x) = x   if x is positive

abs(x) = -x   if x is negative

See also

SGN

Example

This example shows how two different numbers can have the same absolute value.

```
dim i as integer, j as integer
i = 5
j = -5
if abs(i) = abs(j) then
  print "i and j have the same absolute value"
end if
```

ACTIVATE Statement

Used to make *QmodemPro for Windows* the active application on the Windows desktop.

Syntax

```
ACTIVATE
```

Remarks

This command will also restore the *QmodemPro for Windows* window if it is currently minimized.

See also

MINIMIZE, MAXIMIZE, MOVE, SIZE

Example

This example minimizes the *QmodemPro for Windows* window, waits for the string "connect" from the modem, then reactivates the application.

```
minimize
waitfor "connect"
activate
```

ADDENTRY Statement

The ADDENTRY statement is used to add a new dialing directory entry to the phonebook.

Syntax

```
ADDENTRY phoneentry
```

Remarks

The *phoneentry* variable is a variable of type PHONEENTRY, which has this declaration:

```
type phoneentry
  systemname as string*27
  number(1 to 5) as string*19
  userid as string*30
  password as string*25
  notefile as string*79
  scriptfile as string*79
  macrofile as string*79
  icondir as string*79
  emulation as integer
  protocol as integer
  faxonly as integer
  retries as integer
  device as string*20
  connects as long
  uploads as long
  downloads as long
  lastdate as long
  lasttime as long
  iconfile as string*79
  iconid as integer
end type
```

See also

DIAL

Example

This example shows how a new entry might be added to the phonebook.

```
dim entry as phoneentry
entry.systemname = "MSI HQ BBS"
entry.number(1) = "1-805-873-2400"
entry.scriptfile = "msihq.scr"
entry.emulation = ripscrip
entry.protocol = zmodem
```

addentry entry

ADDLFTOCR Function

Used to get or set the current state of the "Add LF to CR" toggle.

Syntax

```
ADDLFTOCR
or
ADDLFTOCR(onoff)
```

Remarks

There are two forms to the ADDLFTOCR function. The first form takes no arguments and simply returns the current setting. The second form sets the state of the toggle and returns the previous state.

In both the parameter and the return value, a nonzero value means line feeds are added to incoming carriage returns.   A zero value means no additional characters are added.

See also

DUPLEX

Example

This example shows how the ADDLFTOCR function can be used in a logon script to turn on the line feed control.

```
dim oldcrlf as integer
oldcrlf = addlftocr(on)
waitfor "UserID"
send "123456"
addlftocr oldcrlf
```

AND Operator

AND performs a bitwise AND operation between its operands.

Syntax

```
op1 AND op2
```

Remarks

op1 and *op2* are logical expressions or numeric values of any integral type (byte, integer, or long integer).

This is the logical truth table for the AND operator:

x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

See also

EQV, IF, IMP, NOT, OR, XOR

Example

The first part of this example shows how the AND operator can be used as a bitwise logical operator to perform manipulations on integers. The second part of this example shows how the AND operator can be used to combine the results of two comparisons and supply the result to an IF statement.

```
dim i as integer
i = 129
if i and 127 = 1 then
  print "yes"
end if

dim j as integer
j = 4
if i = 129 and j = 4 then
  print "yes again"
end if
```

ASC Function

Returns the ASCII value (from 0 to 255) of the first character in a string.

Syntax

```
ASC(string)
```

Remarks

string is any string expression. The return value of this function is the ASCII value of the first character in the string. If the string is empty this function returns zero.

The opposite of the ASC function is the CHR function. The CHR function returns the character corresponding to a particular ASCII code.

See also

CHR, VAL

Example

This example prints 104, which is the ASCII value of "h", the first character in the string "hello".

```
dim a as string
a = "hello"
print asc(a)
```

ATN Function

Returns the arctangent of a numeric expression.

Syntax

```
ATN(number)
```

Remarks

The arctangent of *number* is the angle in radians whose tangent is equal to *number.* You can convert radians to degrees by multiplying by 180/pi (pi is approximately 3.14159).

See also

COS, SIN, TAN

Example

This example uses the atn function to compute the value of pi.

```
print 4*atn(1)
```

AUTOANSWER Statement

Used to send the "Auto answer on" or "Auto answer off" string from the modem configuration.

Syntax

```
AUTOANSWER onoff
```

Remarks

When called with a nonzero argument (ON), this command sends the "Answer on" string from the modem configuration.   When called with a zero argument (OFF), this command sends the "Answer off" string from the modem configuration.

See also

[DIAL](#), [HANGUP](#), [RESETMODEM](#)

Example

This example shows how the AUTOANSWER command might be used to answer an incoming phone call.

```
autoanswer on
loop
until carrier
```

BEEP Statement

Causes *QmodemPro* to sound the regular Windows beep sound.

Syntax

```
BEEP
```

Remarks

If you have attached a .wav file to the "Default Beep" sound in Windows Control Panel, then this command will play the corresponding .wav file.

See also

SOUND

Example

This example simply sounds the Windows default beep.

```
beep
```

BREAK Statement

Sends a break signal to the communications port.

Syntax

```
BREAK
```

Remarks

A break is a special signal sent to the communications port that lasts for a specific amount of time (the time that a break lasts is determined by the **Options/Device/Connection/Break length** setting).

See also

HANGUP

Example

This example sends a break signal, waits for the string "connection lost", then ends the script.

```
break
waitfor "connection lost"
end
```

BYTE Type

Used to declare a variable that can handle byte sized numbers.

Remarks

Variables of byte type can hold values that range from 0 to 255.

See also

<u>DIM</u>, <u>INTEGER</u>, <u>LONG</u>, <u>REAL</u>, <u>TYPE</u>

Example

This example declares a variable of type BYTE and assigns a value to it.

```
dim i as byte
i = 5
print i
```

CALL Statement

Executes a subroutine or function.

Syntax

```
CALL name[( arg [, arg ...])]
or
name [arg[, arg ...]]
```

Remarks

name is the name of the subroutine or function to execute. Subroutines and functions must be at least declared before you can call them (see the DECLARE statement for information on declaring subroutines and functions).

arg is an argument that is passed to the sub-program. Multiple arguments are separated with commas. When using the first syntax with the CALL keyword, parentheses are required around the argument list. When using the second syntax, omitting the CALL keyword, parentheses are not placed around the parameter list.

Arguments are normally passed to the subroutine or function in "reference" mode. This means that if the corresponding argument in the function is changed, the original copy will be changed too. This behavior may be changed by passing an expression to the function, like `a+1` or `(a)` (the parentheses around `a` serve only to create an expression and prevent a subroutine from changing its value). See the example below for an illustration of how this works.

See also

DECLARE, FUNCTION, GOSUB, SUB

Example

This example declares a subroutine called f which takes two arguments. The subroutine adds one to the first argument, and adds two to the second argument. If you run this script, you will notice that the value of a in the main program is changed, while the value of b remains the same. This is because b is not directly passed to the subroutine, but a copy of b is. The subroutine changes the copy and does not affect the actual value of b.

```
declare sub f(x as integer, y as integer)
dim a as integer, b as integer
a = 4
b = 7
call f(a, (b))
print "a is "; a
print "b is "; b

sub f(x as integer, y as integer)
  x = x + 1
  y = y + 2
end sub
```

CAPTURE Statement

Used to open or close a terminal capture file.

Syntax

```
CAPTURE filename
CAPTURE ON
CAPTURE OFF
```

Remarks

filename is the name of the file to which captured data will be appended.

on turns on the capture file specified in **Options/Files/File Definitions**.

off closes any currently open capture file.

If you specify a new capture file while another capture file is still open, the first capture file will be closed before the new one is opened.

If the capture file cannot be opened, the ERR_FILEOPEN error will be generated. This error can be caught with the CATCH statement.

See also

CATCH, PRINTER

Example

This example opens the capture file "test.cap", sends an Enter to the communications port, waits 10 seconds, then closes the capture file.

```
capture "test.cap"
send
delay 10
capture off
```

CARRIER Function

Determines whether the modem is currently on-line and connected to another modem.

Syntax

```
CARRIER
```

Remarks

The CARRIER function returns TRUE if the modem currently reports that it is on-line and connected to another modem. If not, it returns FALSE. The return value of this function corresponds with the state of the "Online/Offline" indicator at the bottom of the terminal window.

See Also

HANGUP

Example

This example sends the string "bye" to the communications port, then waits until the modem reports that carrier is no longer active (that is, the remote modem has hung up).

```
send "bye"
while carrier do
wend
```

CASE Statement

Introduces a new case in a SELECT CASE statement. Please see the description of the SELECT CASE statement for more information and examples.

CATCH Statement

Used to catch runtime errors and perform error recovery actions.

Syntax

```
CATCH errvalue [, errvalue ...]
or
CATCH ALL
```

Remarks

The CATCH mechanism provides a convenient way of responding to errors that may occur while your script is running. The errvalue values must be one of the following:

ERR_ARRAYSUBSCRIPT

Caused by accessing an array using an invalid subscript index.

ERR_FILEOPEN

An error during a file open operation causes this error. The commands that can cause this exception are CAPTURE, OPEN, and SHELL.

ERR_FILERENAME

Caused by an error during a file rename operation. This could be caused by the original file not being found, or open by another application, or the destination filename is already in use. The NAME statement can cause this error.

ERR_FUNCTIONNOTFOUND

Caused by trying to call a function in a dynamic link library (DLL) where the named function does not exist.

ERR_INVALIDFILENUMBER

Caused by using an invalid file number in any file operation, including OPEN, CLOSE, PRINT, INPUT, INKEY, LOF, and so on.

ERR_LIBRARYNOTFOUND

Caused by trying to call a function in a dynamic link library (DLL) where the named DLL does not exist.

ERR_MATH

Caused by trying to divide by zero, taking the logarithm of zero or a negative number, or by taking the square root of a negative number.

ERR_PATH

Caused by an invalid drive or path name in one of the following commands: CHDIR, CHDRIVE, MKDIR, RMDIR.

ERR_TIMEOUT

Caused by a timeout during one of the following commands: INPUT, RECEIVE, WAITFOR.

A CATCH statement may only be placed at the end of a user defined subroutine or function, or at the end of the main program body. During normal operation (the case where no runtime error occurs) the statements after the CATCH statement are skipped. If one of the above runtime errors occurs, the CATCH block for the currently executing function will be searched for a handler for the error. If there is no CATCH block or if there is no specific CATCH handler for the error that occurred, control will return to the function that called the current function. Its CATCH block (if any) will be searched for a handler, and so on up the call chain. If the error propagates all the way up to the main program body and there is no CATCH handler there for the error, then the script is automatically halted with an appropriate error message.

See also

ERROR

Example

This example defines a subroutine that opens a file called "test.dat", reads the first line of text from the file, and closes it. If there is an error opening the file, the subroutine prints an error message.

```
declare sub test
dim a as string
call test(a)
print a

sub test(s as string)
  open "test.dat" for input as #1
  input #1, s
  close #1
catch err_fileopen
  print "could not open test.dat"
end sub
```

CHAIN Statement

Used to transfer control to another script.

Syntax

```
CHAIN scriptname
```

Remarks

This command terminates execution of the current script and starts execution of the named script. All currently open files are closed and variables are discarded.

The script named in this command must be the name of a compiled script with the proper .scc extension. Scripts are not automatically compiled by this command.

See also

END, STOP

Example

This example shows two files, a.scr and b.scr, and demonstrates how you can start execution of the second script from the first.

```
[in file a.scr]
print "in a.scr"
chain "b.scc"

[in file b.scr]
print "in b.scr"
```

CHDIR Statement

Changes the current directory.

Syntax

```
CHDIR directory
```

Remarks

This command changes to the directory named in the *directory* argument.

As with the DOS CHDIR command, this does not change the current drive, even if it is specified in the argument. However, if you change the current directory for a drive that is not the current drive, the change will be remembered until you next change to the new drive using the CHDRIVE command.

See also

CHDRIVE, CURDIR, CURDRIVE, MKDIR, RMDIR

Example

This example shows various ways the CHDIR command can be used.

```
chdir "\"           'change to the root directory
chdir "c:\qmwin"    'change to C:\QMWIN
dim newdir as string
newdir = "c:\temp"  'assign specified directory to
                    'string variable NEWDIR
chdir newdir        'change to the new directory
```

CHDRIVE Statement

Changes the current drive.

Syntax

```
CHDRIVE driveletter
```

Remarks

This command changes the current drive to *driveletter*. If *driveletter* is longer than one character, only the first character in the string is used.

See also

CHDIR, CURDIR, CURDRIVE

Example

This example takes advantage of the fact that only the first letter of the argument to CHDRIVE is used, and changes to the directory "c:\test" no matter which drive this script was started from.

```
dim a as string
a = "c:\test"
chdrive a
chdir a
```

CHR Function

Returns the ASCII character corresponding to the specified ASCII code value in the range of 0 to 255.

Syntax

```
CHR(number)
```

Remarks

This function returns the ASCII character corresponding to *number.*

The opposite of the CHR function is the ASC function. The ASC function returns the ASCII character value of the first character in a string.

See also

ASC, SPACE, STRING

Example

This example prints "Hi" using the ASCII values of the characters "H" and "i".

```
print chr(72); chr(105)
```

CLOSE Statement

Closes a file or files opened with the OPEN statement.

Syntax

```
CLOSE [[#]filenumber[, [#]filenumber]...]
```

Remarks

filenumber is the number of an open file. The file associated with the given file number will be closed.

CLOSE without any parameters will close all open files.

Although *QmodemPro for Windows* will automatically close files when the script terminates, you should always close any files that you open. There are only a limited number of file numbers available; if you run out of file numbers you will not be able to open any more files.

See also

OPEN, RESET

Example

This file opens a new file called "test.txt" for output, writes a line of test data to the file, then closes the file.

```
open "test.txt" for output as #1
print #1, "test data"
close #1
```

CLS Statement

Clears the terminal screen and returns the cursor to the top left corner. This command also resets the current text color to the default text color.

Syntax

```
CLS
```

Remarks

This command is similar to the DOS CLS "clear screen" command.

See also

COLOR

Example

This example clears the screen and writes "Hello, world!" in the upper left hand corner.

```
cls
print "Hello, world!"
```

COLOR Statement

Sets the current terminal color to the specified foreground and background.

Syntax

```
COLOR foreground [, background]
```

Remarks

Sets the current terminal color to the specified foreground and background values. If the background value is not specified, then it remains unchanged.

The color values used by this command are as follows:

0   Black

1   Maroon

2   Green

3   Olive

4   Navy

5   Purple

6   Teal

7   Gray

8   Silver

9   Red

10   Lime

11   Yellow

12   Blue

13   Fuschia

14   Aqua

15   White

See also

CLS, SCREEN

Example

This example sets the color to yellow on blue and prints a text string in that color.

```
color 11, 4
```

```
print "yellow text on blue background"
```

CONFIGCAPTUREFILE Function

Syntax

    CONFIGCAPTUREFILE

Remarks

This function returns the exact text contained in the Options/Files/File definitions dialog entry for the capture file name. This is usually the fully qualified path and filename of the capture file, but may return only the filename if no path was entered in the config dialog.

See also

CONFIGDOWNLOADPATH, CONFIGLOGFILE, CONFIGSCRIPTPATH, CONFIGSCROLLBACKFILE, CONFIGTRAPFILE, CONFIGUPLOADPATH

Example

    UPLOAD (CONFIGCAPTUREFILE, Zmodem)

CONFIGDOWNLOADPATH Function

Syntax

```
CONFIGDOWNLOADPATH
```

Remarks

This function returns the exact text contained in the Options/Files/Path definitions dialog entry for the download path. This is the fully qualified path for the download directory, without a trailing backslash.

See also

[CONFIGCAPTUREFILE](#), [CONFIGLOGFILE](#), [CONFIGSCRIPTPATH](#), [CONFIGSCROLLBACKFILE](#), [CONFIGTRAPFILE](#), [CONFIGUPLOADPATH](#)

Example

```
CHDIR CONFIGDOWNLOADPATH
```

CONFIGLOGFILE Function

Syntax

```
CONFIGLOGFILE
```

Remarks

The function returns the exact text contained in the Options/Files/File definitions dialog entry for the logfile name. This is usually the fully qualified path and filename of the log file, but may return only the filename if no path was entered in the config dialog.

See also

CONFIGDOWNLOADPATH, CONFIGCAPTUREFILE, CONFIGSCRIPTPATH, CONFIGSCROLLBACKFILE, CONFIGTRAPFILE, CONFIGUPLOADPATH

Example

```
UPLOAD (CONFIGLOGFILE, Zmodem)
```

CONFIGSCRIPTPATH Function

Syntax

```
CONFIGSCRIPTPATH
```

Remarks

The function returns the exact text contained in the Options/Files/Path definitions dialog entry for the scripts path. This is the fully qualified path for the scripts directory, with no trailing backslash.

See also

CONFIGDOWNLOADPATH, CONFIGCAPTUREFILE, CONFIGLOGFILE, CONFIGSCROLLBACKFILE, CONFIGTRAPFILE, CONFIGUPLOADPATH

Examples

```
CHDIR CONFIGSCRIPTPATH
```

CONFIGSCROLLBACKFILE Function

Syntax

```
CONFIGSCROLLBACKFILE
```

Remarks

The function returns the exact text contained in the Options/Files/File definitions dialog entry for the scrollback filename. This is usually the fully qualified path and filename of the scrollback file, but may return only the filename if no path was entered in the config dialog.

See also

CONFIGDOWNLOADPATH, CONFIGLOGFILE, CONFIGSCRIPTPATH, CONFIGCAPTUREFILE, CONFIGTRAPFILE, CONFIGUPLOADPATH

Example

```
UPLOAD CONFIGSCROLLBACKFILE, Zmodem)
```

CONFIGTRAPFILE Function

Syntax

```
CONFIGTRAPFILE
```

Remarks

The function returns the exact text contained in the Options/Files/File definitions dialog entry for the trap filename. This is usually the fully qualified path and filename of the trap file, but may return only the filename if no path was entered in the config dialog.

See also

CONFIGDOWNLOADPATH, CONFIGLOGFILE, CONFIGSCRIPTPATH, CONFIGSCROLLBACKFILE, CONFIGCAPTUREFILE, CONFIGUPLOADPATH

Example

```
UPLOAD CONFIGTRAPFILE, Zmodem)
```

CONFIGUPLOADPATH Function

Syntax

```
CONFIGUPLOADPATH
```

Remarks

The function returns the exact text contained in the Options/Files/File definitions dialog entry for the upload file area path. This is the fully qualified path for the upload path, with no trailing backslash.

See also

CONFIGDOWNLOADPATH, CONFIGLOGFILE, CONFIGSCRIPTPATH, CONFIGSCROLLBACKFILE, CONFIGTRAPFILE, CONFIGCAPTUREFILE

Examples

```
CHDIR CONFIGUPLOADPATH
```

CONST Statement

Used to assign symbolic names that will be used in place of actual values.

Syntax

```
CONST name = expression [, name = expression ...]
```

Remarks

name is the name of the new constant.

expression is the value to assign to the symbolic constant name.

The type of name is determined by the type of the expression.

Constants defined in subroutines or functions can only be used within the subroutine or function. Constants defined with a CONST statement in the main program body can be used throughout the program.

Example

This example declares the constant "myname" and assigns a value to it.

```
const myname = "John Doe"
print "My name is "; myname
```

COS Function

Returns the cosine of an angle.

Syntax

```
COS(angle)
```

Remarks

angle is the measurement of an angle expressed in radians. You can convert radians to degrees by multiplying by 180/pi (pi is approximately 3.14159).

See also

ATN, SIN, TAN

Example

This example prints the cosine of 1 radian.

```
print cos(1)
```

CSRLIN Function

Returns the current vertical coordinate position (row number) of the cursor.

Syntax

```
CSRLIN
```

Remarks

This value is usually an integer in the range 1 through 25, but may be larger depending on the number of lines set in the **Options/Emulations** dialog.

See also

LOCATE, POS

Example

This example clears the screen, then prints the cursor line twice. The first time it will be 1 since the cursor is on the top line of the screen, and the second time it will be 2 (the cursor moved down because of the first print statement).

```
cls
print csrlin
print csrlin
```

CURDIR Function

Returns the current drive and directory.

Syntax

```
CURDIR
```

Remarks

The current directory of the current drive is returned without a trailing backslash (unless the current directory is the root directory).

See also

CHDIR, CHDRIVE, CURDRIVE, MKDIR, RMDIR

Example

This example prints the current directory for the current drive.

```
print "The current directory is "; curdir
```

CURDRIVE Function

Returns the current drive letter.

Syntax

```
CURDRIVE
```

Remarks

The current drive letter is returned as an uppercase letter. This is the same drive that is returned by the CURDIR function.

See also

CHDIR, CHDRIVE, CURDIR, MKDIR, RMDIR

Example

This example prints the current drive letter.

```
print "The current drive is "; curdrive
```

DATE Function

Returns the current date as a string.

Syntax

```
DATE
```

Remarks

The date is returned in the format specified in Windows Control Panel, International section, short date format.

See also

TIME

Example

This example prints today's date.

```
print "Today is "; date
```

DECLARE Statement

Allows you to declare subroutines and functions before their actual definition. Also allows declaration of DLL subroutines and functions.

Syntax

```
DECLARE [FUNCTION | SUB] name [LIB "libname" [ALIAS "aliasname"]]
[(argument list)] [AS returntype]
```

Remarks

name is the name of the subroutine or function.

The argument list lists the parameters to the subroutine or function. This argument list must match the argument list declared in the actual definition of the subroutine or function.

The LIB and ALIAS clauses are used to declare a function that actually exists in another DLL. For more information on declaring and using DLL functions, see Using DLL Functions.

The final AS clause is used to declare the return type of a function.

Functions must be declared before they can be used. If the function definition appears later in the script source file from where you want to call the function, the DECLARE statement can be used to declare the function before the call.

See also

CALL, FUNCTION, SUB

Example

This example uses the DECLARE statement to declare a function so it can be used before its actual definition appears later in the file.

```
declare function timestwo(x as integer)
print timestwo(5)

function timestwo(x as integer)
  timestwo = x * 2
end function
```

DEL Statement

Deletes a file from disk.

Syntax

```
DEL filename
```

Remarks

filename is the name of the file to delete. DOS wildcards * and ? are not supported.

This statement is identical to the script KILL command.

You cannot delete an open file, whether it is has been opened by your script, *QmodemPro for Windows*, or another application.

See also

KILL, RMDIR

Example

This example deletes the file "test.dat" from the current directory.

```
del "test.dat"
```

DELAY Statement

Used to suspend script execution for a certain time interval.

Syntax

```
DELAY time
```

Remarks

time is the amount of time to suspend execution, expressed in seconds. If you want to delay for less than a second, use ordinary decimal notation.

This command is identical to the script PAUSE command.

See also

PAUSE, WAITFOR, WHEN QUIET, WHEN TIME

Example

This example uses the "atz" command to reset a Hayes-compatible modem, waits for half a second, then sends a command to dial a telephone number.

```
send "atz"
delay 0.5
send "atdt5551212"
```

DEVICE Statement

Changes the currently active device.

Syntax

```
DEVICE devicename
```

Remarks

The device list is searched for *devicename*. If a matching device is found it is activated. If no matching device is found then the current device remains unchanged.

The special device name "-Default-" switches to the default device, and the device name "-Current-" switches to the current device (switching to the current device only serves to reinitialize the communications port).

See also

DIAL, SETCOMM

Examples

This example switches to the device "USR HST 14400" and sends a command to initialize the modem.

```
device "USR HST 14400"
send "atz"
```

DIAL Statement

This statement is used to dial an entry or entries from the phonebook.

Syntax

```
DIAL ENTRY number
or
DIAL GROUP groupname
or
DIAL SEARCH string
or
DIAL MANUAL number
```

Remarks

There are four forms to the DIAL command:

DIAL ENTRY

This form allows you to dial a specific entry number in the phonebook.

DIAL GROUP

This form allows you to dial all the entries in a named group.

DIAL SEARCH

This form allows you to search for a string and dial all entries that contain that string.

DIAL MANUAL

This form allows you to dial a specific phone number from the script.

After dialing a number from a script in this way, the script file specified in the dialing directory (if any) is not executed. After connect the script proceeds from the next statement after the DIAL command.

See also

ADDENTRY, DIALNEXT, HANGUP, LASTCONNECTPASSWORD, LASTCONNECTUSERID

Examples

This example shows each of the three ways the DIAL command can be used.

```
dial entry 3
dial group "Morning mail"
dial search "Mustang"
dial manual "555-1212"
```

DIALNEXT Function

This function is used to dial the next entry in a group after using the DIAL GROUP command.

Syntax

```
DIALNEXT
```

Remarks

When using the DIAL GROUP command, all the entries in the phonebook corresponding to the dialed group are marked for dial.   After connecting to an entry, use the DIALNEXT function to continue dialing the remaining marked entries in the phonebook.

This function returns FALSE if there are no further marked entries in the phonebook, otherwise it returns TRUE.

See also

ADDENTRY, DIAL, HANGUP

Examples

This example shows how the DIALNEXT command is used in conjunction with DIAL GROUP.

```
dial group "Morning mail"
do
  ... do whatever needs to be done online ...
loop while dialnext
```

DIALOG Statement

This statement is used to declare a dialog box type.

Syntax

```
DIALOG dialogtype x, y, w, h
  [CAPTION caption]
  [FONT size, fontname]
  [integer-field AS CHECKBOX title, id, x, y, w, h]
  [integer-field AS COMBOBOX id, x, y, w, h]
  [CTEXT title, id, x, y, w, h]
  [DEFPUSHBUTTON title, id, x, y, w, h]
  [string-field AS EDITTEXT id, x, y, w, h]
  [GROUPBOX title, id, x, y, w, h]
  [integer-field AS LISTBOX id, x, y, w, h]
  [LTEXT title, id, x, y, w, h]
  [PUSHBUTTON title, id, x, y, w, h]
  [integer-field AS RADIOBUTTON title, id, x, y, w, h]
  [RTEXT title, id, x, y, w, h]
  ...
END DIALOG
```

Remarks

The DIALOG statement declares a *dialog template*. A dialog template is much like a user defined type in that it contains named fields in which you can place information. Dialog templates can also have a number of unnamed fields which serve to place extra information in the Windows dialog that is created based on this template.

Dialog boxes are discussed in depth in the section titled Using Dialog Boxes.

See also

DIALOGBOX

DIALOGBOX Function

Displays and executes a dialog box based on a dialog box template.

Syntax

```
DIALOGBOX(dialogvar)
```

Remarks

This function creates and executes a dialog box based on the dialog box variable. Its return value depends on the event that caused the dialog box to close. In most cases this will be either IDOK or IDCANCEL depending on whether the user pressed the OK button or the Cancel button to close the dialog box. In more advanced cases the meaning of the return value will be user defined.

The dialog box variable used in this function should not be already active.

Dialog boxes are discussed in depth at the end of this chapter, in   the section titled Using Dialog Boxes.

See also

DIALOG

DIM Statement

The DIM statement is used to declare variables of any type including array types. An array is a variable containing a series of values that are all of the same type.

Syntax

```
DIM var[([lowerbound TO] upperbound)] AS type[, var[([lowerbound TO]
upperbound)] AS type...]
```

Remarks

var is the name of the array or variable being declared.

lowerbound and upperbound declare the lowest and highest subscript values that are allowed if an array is being declared. If lowerbound is omitted, it defaults to zero.

Each variable must have an associated type declaration with the AS clause.

Variables declared within a subroutine or function are only available from within that subroutine or function. Variables declared in the main program body are available throughout the entire script file.

An array consists of a number individual variables, called "elements", which are referred to by numbers, called "subscripts" indicating the position of each element in the array.

See also

STATIC

Example

This example declares a simple integer "i" and an array "a" which refers to a sequence of six integers, numbered a(0) through a(5).

```
dim i as integer, a(5) as integer
for i = 1 to 5
  a(i) = i*2
next
for i = 1 to 5
  print a(i)
next
```

DO ... LOOP Statement

Repeatedly executes a block of statements while (or until) a specified condition is met.

Syntax

```
DO [{WHILE | UNTIL} expression]
  [statements]
LOOP
```

or

```
DO
  [statements]
LOOP [{WHILE | UNTIL} expression]
```

Remarks

The first example above tests for the specified condition at the beginning of the loop, and exits the loop when the condition is met.

The second example tests for the specified condition at the end of the loop, and continues until the condition is met.

expression is any logical expression that evaluates to either true (nonzero) or false (zero). The keyword WHILE repeats the loop while the expression remains true. The UNTIL keyword repeats the loop until the expression becomes true.

statements are program statements that are repeated. Note that statements are optional; a loop can be empty, waiting for an external event such as a keypress or an incoming character.

Every DO statement in a program must have a corresponding LOOP, and each LOOP must have a preceding DO.

See also

EXIT DO, FOR ... NEXT, WHILE ... WEND

Example

This example prints out the integers from 1 through 5 using a DO WHILE ... LOOP statement.

```
dim i as integer
i = 0
do while i < 5
  i = i + 1
  print i
loop
```

DOORWAY Function

Used to get or set the current state of the Doorway toggle.

Syntax

```
DOORWAY
or
DOORWAY(onoff)
```

Remarks

There are two forms to the DOORWAY function. The first form takes no arguments and simply returns the current setting. The second form sets the state of the toggle and returns the previous state.

In both the parameter and the return value, a nonzero value means Doorway mode is turned on.   A zero value means Doorway mode is turned off.

When turned on, Doorway mode causes all keystrokes that can be typed on the keyboard to be sent directly to the remote.   This will work only with a remote host that understands the Doorway keystrokes.

Example

This example shows how to save the doorway setting, change it, and then restore it..

```
dim olddoorway as integer
olddoorway = doorway(on)
  ...
doorway olddoorway
```

DOSVERSION Function

Returns the current DOS version as a string.

Syntax

```
DOSVERSION
```

Remarks

This function returns the DOS version as a string in the format X.Y. X is the major version number and Y is the minor revision number.

For example, DOS version 6.0 would be reported as "6.0".

See also

<u>VERSION</u>, <u>WINVERSION</u>

Example

```
print "Currently using DOS "; dosversion
```

DOWNLOAD Function

Used to receive files from a remote computer.

Syntax

```
DOWNLOAD(filename, protocol)
```

Remarks

This function initiates a file transfer to receive files from a remote computer. At the time this command is executed, the remote computer must already have started the file transfer. If you are connecting to a bulletin board system (BBS) then your script should already have sent the command to the BBS that will start the file transfer.

The protocol is one of the following predefined constants:

ASCII, XMODEM, XMODEMCRC, XMODEM1K, XMODEM1KG, YMODEM, YMODEMG, ZMODEM, KERMIT, BPLUS

For the first five protocols, the filename parameter must specify an actual file name in which to place the received file. The ASCII and Xmodem variant protocols do not supply a filename, so one must be supplied in the DOWNLOAD function.

For the last five protocols, the filename parameter should be the name of a directory in which to place the received files. If this parameter is an empty string ("") then the download directory specified in **Options/Files/Path Definitions** will be used.

The DOWNLOAD function returns zero if the file transfer was successful. If the transfer was unsuccessful, DOWNLOAD returns the error code describing the error. For a list of error codes see Chapter 4.

This function is identical to the RECEIVEFILE function.

See also

RECEIVEFILE, SENDFILE, UPLOAD

Example

This example receives a file from a remote computer, assuming that the remote computer has already started to send the file.

```
if download("", Zmodem) = 0 then
  print "file transfer ok!"
end if
```

DUPLEX Function

Used to get or set the current duplex (local echo) setting.

Syntax

```
DUPLEX
or
DUPLEX(onoff)
```

Remarks

There are two forms to the DUPLEX function. The first form takes no arguments and simply returns the current duplex setting. The second form sets the duplex and returns the previous state of the duplex setting.

In both the parameter and the return value, a nonzero value means full duplex or no local echo. A zero value means half duplex or local echo.

See also

HOSTECHO

Example

This example shows how the DUPLEX function can be used to control the local echo of characters sent to the communications port.

```
dim oldduplex as integer
oldduplex = duplex(0)
send "this will be echoed locally"
duplex oldduplex
send "this will not be echoed locally"
```

EDITFILE Statement

Used to invoke the internal editor with a specified file.

Syntax

```
EDITFILE filename
```

Remarks

This function brings up the internal editor with the named file loaded ready for editing. Note that that script continues to run after the editor is started.

See also

[VIEWFILE](#)

Example

This example shows how the EDITFILE command might be used to edit the host mode user file.

```
editfile "host.usr"
```

ELSE Statement

The ELSE statement is used to introduce the alternative portion of an IF statement. See the discussion of the IF statement for more information.

ELSEIF Statement

The ELSEIF statement is used to introduce an optional alternative portion of an IF statement. See the discussion of the IF statement for more information.

EMULATION Statement

This statement is used to change the current terminal emulation.

Syntax

```
EMULATION emulation
```

Remarks

The emulation argument must be one of the following predefined constants:

ADDSVP60, ADM3A, ANSI, AVATAR, DEBUGASCII, DEBUGHEX, DG100, DG200, DG210, HAZELTINE1500, HEATH19, IBM3101, RIPSCRIP, TTY, TVI910, TVI912, TVI920, TVI922, TVI925, TVI950, TVI955, VT100, VT102, VT220, VT320, VT52, WYSE30, WYSE50, WYSE60, WYSE75, WYSE85, WYSE100, WYSE185

See also

DEVICE, DIAL

Example

This example changes the current terminal emulation to RIPscrip.

```
emulation ripscrip
```

END Statement

Ends a script and returns to normal terminal operation.

Syntax

```
END
```

Remarks

The END statement immediately terminates the currently executing script and returns to normal terminal operation. It may be executed at any time. Any currently open files are closed before the script ends.

See also

STOP

Example

This example print a string to the terminal, then ends the script. The second print statement is never executed.

```
print "hello world"
end
print "this is never executed"
```

ENVIRON Function

Returns the value of the specified DOS environment variable.

Syntax

```
ENVIRON(envname)
```

Remarks

envname is the name of the environment variable to retrieve. The *envname* parameter should be in upper case. If the specified environment variable is not found, an empty string will be returned.

Example

This example prints out the current DOS PATH setting.

```
print environ("PATH")
```

EOF Function

Used to determine whether the end of a file has been reached.

Syntax

```
EOF(filenum)
```

Remarks

This function returns TRUE if the end of the file specified by the *filenum* parameter has been reached.

See also

LOC, LOF, SEEK

Example

This example prints the contents of a text file line by line until the end of the file is encountered.

```
dim a as string
open "test.dat" for input as #1
do while not eof(1)
  input #1, a
  print a
loop
close #1
```

EQV Operator

EQV performs a bitwise equivalence operation between its operands.

Syntax

```
op1 EQV op2
```

Remarks

op1 and op2 are logical expressions or numeric values of any integral type (byte, integer, or long integer).

This is the logical truth table for the EQV operator:

a	b	a EQV b
0	0	1
0	1	0
1	0	0
1	1	1

See also

AND, IMP, NOT, OR, XOR

Example

This example shows how the EQV operator can be used in an IF statement.

```
dim i as integer, j as integer, k as integer
i = 10
j = 8
k = 6
if i > j eqv j > k then
  print "both expressions are true or both are false"
end if
```

ERROR Statement

Causes a predefined or user defined runtime error to occur.

Syntax

```
ERROR errvalue
```

Remarks

If there is a CATCH handler for the error value, then it will be executed. Otherwise, the script will be terminated automatically. See the CATCH statement for a list of predefined runtime error values.

See also

CATCH

Example

This example forces the ERR_FILEOPEN error to occur, which will cause the script to jump immediately to the handler for ERR_FILEOPEN. As a result, the first print statement won't be executed.

```
error err_fileopen
print "this won't be printed"

catch err_fileopen
print "caught the error"
```

EXISTS Function

Determines whether a specific file exists on disk.

Syntax

```
EXISTS(filename)
```

Remarks

This function returns TRUE if the specified file exists on disk. DOS wildcards are not accepted.

See also

FINDFIRST, FINDNEXT

Example

This example determines whether the "c:\autoexec.bat" file still exists.

```
if exists("c:\autoexec.bat") then
  print "autoexec.bat is still there"
end if
```

EXIT Statement

Aborts a loop, subroutine, or procedure without waiting for normal termination or return.

Syntax

```
EXIT [DO | FOR | FUNCTION | SUB]
```

Remarks

One of the four keywords above must be included in the EXIT statement.

Use the EXIT DO and EXIT FOR statements to exit a loop defined by a DO ... LOOP or a FOR ... NEXT statement.

If a loop is nested, the EXIT statement will stop only the current (innermost) loop.

Use the EXIT FUNCTION and EXIT SUB statements to abort procedures defined by a FUNCTION or SUB statement.

If you are exiting a function with EXIT FUNCTION and have not already assigned a value to the function result, the function will return zero or an empty string, depending on the type of the function.

See also

DO ... LOOP, FOR ... NEXT, FUNCTION, SUB

Example

This example uses the EXIT DO statement to break out of a loop when a certain condition is satisfied.

```
dim i as integer
i = 0
do
  i = i + 1
  if i > 10 then exit do
  print i
loop
```

EXP Function

Returns the natural antilogarithm of a value.

Syntax

```
EXP(number)
```

Remarks

This function returns e (the natural logarithm base) raised to the power of *number*.

See also

[LOG](LOG)

Example

This example raises e to the power of 2 (in other words, it computes the square of e) and prints the result.

```
print exp(2)
```

FINDFIRST Function

Used to find the first of a set of files matching a wildcard filename specification.

Syntax

```
FINDFIRST(name, attr, sr)
```

Remarks

name is the filename specification to search for, and can include DOS wildcard characters * and ?.

attr is the file attribute to look for. To search for normal files use 0 for this parameter.

sr is a variable of type SEARCHREC that will hold the file information for the file found.

This function is normally used in conjunction with the FINDNEXT function to gather a list of files.

The declaration for the SEARCHREC type is as follows:

```
type searchrec
  attribute as byte
  time as integer
  date as integer
  size as long
  name as string*13
end type
```

The date and time are encoded using the following formula:

```
time = (hours * 2048) + (minutes * 32) + (seconds / 2)
date = ((year - 1980) * 512) + (month * 32) + day
```

This function returns zero if a file matching the specification was found. It returns a nonzero value if no files were found.

See also

EXISTS, FINDNEXT

Example

This example lists all the files in the "c:\qmwin" directory.

```
dim sr as searchrec, r as integer
r = findfirst("c:\qmwin\*.*", 0, sr)
do while r = 0
  print sr.name
  r = findnext(sr)
loop
```

FINDNEXT Function

Finds the next of a set of files that was started using the FINDFIRST function.

Syntax

```
FINDNEXT(sr)
```

Remarks

sr is a variable of type SEARCHREC.

This function is always used in conjunction with the FINDFIRST function to search for a list of files.

This function returns zero if another file matching the specification was found. It returns a nonzero value if no more files were found.

See also

EXISTS, FINDFIRST

Example

See the FINDFIRST function for an example of how to use this function.

FIX Function

Returns the integer or whole number portion for a numeric expression.

Syntax

```
FIX(number)
```

Remarks

FIX and INT operate the same for positive values, however with negative values their operation is slightly different. FIX returns the next-higher integer with negative values, while INT returns the next lower integer with negative values.

See also

INT

Example

This example shows how the FIX function rounds negative numbers up to the next higher integer. In this example the output will be 3 and 4.

```
print fix(3.7), fix(-4.9)
```

FLUSH Statement

Flushes either or both of the input and output communications buffers.

Syntax

```
FLUSH [INPUT] [OUTPUT]
```

Remarks

The INPUT keyword indicates that the communications input buffer should be flushed.

The OUTPUT keyword indicates that the communications output buffer should be flushed.

See also

BREAK, HANGUP

Example

This example shows how the FLUSH INPUT statement might be used to get rid of stray input data caused by hanging up the modem.

```
hangup
delay 2
flush input
```

FOR ... NEXT Statement

Executes a sequence of program statements a specified number of times.

Syntax

```
FOR index = initial TO final [STEP step]
   [statements]
NEXT [index]
```

Remarks

index is a numeric variable of type byte, integer, long, or real, and counts the number of times the loop executes.

initial is the initial value assigned to the counter at the beginning of the loop.

final is the ending value against which the loop tests the variable index for each pass.

step is the amount by which the counter is incremented after each pass in a FOR ... NEXT loop. The default value is 1.

When the value of index is higher than the value of final, the loop finishes, and returns control to the statement following the keyword NEXT. If the step value is less than zero, then the index is compared against the final value and the loop is stopped if *index* is less than *final*.

See also

DO ... LOOP, EXIT, WHILE ... WEND

Example

This example prints "hello" ten times.

```
dim i as integer
for i = 1 to 10
  print "hello "; i
next
```

FREEFILE Function

Returns the next available file number.

Syntax

```
FREEFILE
```

Remarks

You can assign file numbers on the fly by storing the FREEFILE result in a variable and passing the variable to OPEN and CLOSE statements.

Return value

This function returns the first available file number (one that does not refer to a currently open file). If no file number is available, then the return value is 1.

See also

OPEN

Example

This example opens a file and outputs some test data to it. It doesn't matter how many files are currently open because it uses the FREEFILE function to find a free file number.

```
dim f as integer
f = freefile
open "test.dat" for output as #f
print #f, "this is a test"
close #f
```

FUNCTION Statement

Allows you to define your own subprograms that return a value to the caller.

Syntax

```
FUNCTION name[(arg AS type[, arg AS type]...)] AS type
  ...
  name = expr
  ...
END FUNCTION
```

Remarks

name is the name you assign to the function.

arg is the name of a formal argument to the function. Each argument must have an associated type declaration using the AS keyword.

You may define local variables within your user-defined function. You can use STATIC declarations to preserve the value of individual variables across function calls.

To return a value from the function, use an assignment to the name of the function. The value you assign to the function is remembered until the function ends, at which point it is retrieved and returned to the caller of the function.

Functions cannot be defined within another function definition.

See also

DECLARE, EXIT, STATIC, SUB

Example

This example declares a function that returns its argument multiplied by two and incremented by one.

```
function f(x as integer) as integer
  f = x*2 + 1
end function

print f(2)
print f(f(7))
```

GET Statement

Reads information from a random-access or binary file into a record variable.

Syntax

```
GET [#]filenumber, [position], variable
```

Remarks

filenumber is the number assigned to an open file.

position is the number of the record in a random access file, or the number of the byte in a binary file. Note that unlike some other languages, the first record or byte in the file is number 1, not number 0. If the position is not specified, then the record is read from the current file position.

variable is the name of the variable that receives the returned data. WARNING: If you are reading a record from a random access file, you must ensure that the variable is of the correct type, otherwise unpredictable results may occur.

See also

INPUT, OPEN, PUT, TYPE ... END TYPE

Example

This example creates a random access file, writes a record to it from the variable d, and reads it back into the variable z. It then prints out the contents of z to make sure that the operation succeeded.

```
type daterec
  day as integer
  month as integer
  year as integer
end type
dim d as daterec, z as daterec
d.day = 11
d.month = 10
d.year = 1993
open "test.dat" for random as #1 len = len(daterec)
put #1, 1, d
get #1, 1, z
close #1
print z.day, z.month, z.year
```

GOSUB ... RETURN Statement

Branches unconditionally to the specified line number or label to execute a subroutine.

Syntax

```
GOSUB {line|label}
        .
label:
        .
RETURN
```

Remarks

line and *label* indicate the line number or label at which the subroutine should begin executing.

The RETURN statement resumes execution at the statement following the original GOSUB statement.

You cannot use GOSUB to branch into or out of a sub-program or user-defined function.

The SUB and FUNCTION constructs provide a much better method of supporting subprograms than GOSUB and RETURN.

See also

FUNCTION, GOTO, SUB

Example

This example has a very simple subroutine called "printit" that prints the value of i and returns.

```
dim i as integer
i = 5
gosub printit
i = 9
gosub printit
end
printit:
print i
return
```

GOTO Statement

Branches unconditionally to a specified line or label.

Syntax

```
GOTO {line|label}
```

Remarks

Program execution continues at the referenced line or label, or the first executable statement immediately following the line or label.

You cannot use GOTO to branch into or out of a sub-program or user-defined function.

Proponents of structured programming recommend against using GOTO statements wherever IF statements and DO loops can accomplish the same task, on the grounds that structured code is more efficient and easier to maintain.

See also

FUNCTION, GOSUB, SUB

Example

This example demonstrates a simple loop structure using a GOTO statement.

```
dim i as integer
i = 0
again:
i = i + 1
if i < 10 then goto again
```

HANGUP Statement

Hangs up the modem by sending the modem hangup string, without asking for confirmation.

Syntax

```
HANGUP
```

Remarks

This statement sends the modem hangup string as defined in **Options/Device/Modem**. If the hangup string indicates that DTR be dropped or a BREAK sent, this command will do so.

See also

BREAK, CARRIER

Example

This example sends a command to log off an on-line service, waits five seconds, then hangs up the modem.

```
send "bye"
delay 5
hangup
```

HEX Function

Converts an integer or long integer expression to a hexadecimal (base 16) value.

Syntax

```
HEX(number)
```

Remarks

number is the numeric variable or expression that HEX converts. The value may be any numeric type -- it will be converted to an integer or a long integer. Fractional values are ignored.

See also

OCT

Example

This example prints the hexadecimal representation of the number 42.

```
print hex(42)
```

HOSTECHO Function

Used to turn on and off character echo conventions that are used with the host mode.

Syntax

```
HOSTECHO [ON | OFF]
```

Remarks

This command modifies the operation of *QmodemPro* in the following ways:

Incoming data is not displayed on the screen, written to the capture file, or sent to the printer. It is only sent to the script program for processing.

Data that is output to the communications port is automatically echoed to the terminal screen.

When using the SEND command to output data to the communications port, a carriage return/line feed pair is sent at the end of the line if no semicolon is present (this is in contrast to the standard behavior of sending just a carriage return).

See also

DUPLEX

Example

This example shows what will happen when HOSTECHO is turned on. For the best example of how this command works, see the host mode script HOST.SCR.

```
dim name as string
hostecho on
send "What is your first name? ";
```

IF Statement

Makes a decision regarding program flow, based on the result returned by an expression.

Syntax

```
IF expression THEN then-part ELSE else-part
or
IF expression1 THEN
        statements-1
[ELSEIF expression2 THEN
        statements-2]
[ELSE
        statements-n]
END IF
```

Remarks

expression is an expression that yields a result of true (nonzero) or false (zero).

then-part is a set of statements to be executed if *expression* is true.

else-part is a set of statements that will be executed if *expression* is false.

ELSE and ELSEIF are optional. ELSE allows you to execute a set of statements if *expression* is false.

ELSEIF allows you to test for several different conditions within a single IF statement. In the construct above, *expression2* would only be tested if *expression1* was false.

See also

DO ... LOOP, SELECT CASE

Example

This example demonstrates both types of IF statement syntax.

```
dim i as integer
i = 5
if i = 7 then print "is seven" else print "not seven"
if i = 5 then
  print "i is 5"
elseif i = 9 then
  print "i is 9"
else
  print "i is neither 7 nor 9"
end if
```

IMP Operator

IMP performs a bitwise implication operation between its operands.

Syntax

```
op1 IMP op2
```

Remarks

op1 and *op2* are logical expressions or numeric values of any integral type (byte, integer, or long integer).

This is the logical truth table for the IMP operator:

a	b	a IMP b
0	0	1
0	1	1
1	0	0
1	1	1

See also

AND, EQV, NOT, OR, XOR

Example

This example prints out the truth table above.

```
dim a as integer, b as integer
print "a", "b", "a IMP b"
for a = 0 to 1
  for b = 0 to 1
    print a, b, a IMP b
  next
next
```

INKEY Function

Reads a character from the keyboard, communications port, or a file.

Syntax

```
INKEY(filenumber)
```

Remarks

If there is no character ready when reading from the keyboard or communications port, this function returns an empty string.

When reading from a file or communications port, or if there is an ordinary character waiting from the keyboard, this function returns a single-character string containing the character.

When reading from the keyboard, the following strings are returned if the corresponding special key is pressed:

"Tab", "F1", "F2", "F3", "F4", "F5", "F6", "F7", "F8", "F9", "F10", "F11", "F12", "Backspace", "Enter", "Scroll Lock", "Pause", "Gray Insert", "Gray Delete", "Gray Home", "Gray End", "Gray Pageup", "Gray Pagedown", "Gray Up", "Gray Down", "Gray Left", "Gray Right", "Numlock", "Pad /", "Pad *", "Pad -", "Pad +", "Pad Enter", "Pad .", "Pad 0", "Pad 1", "Pad 2", "Pad 3", "Pad 4", "Pad 5", "Pad 6", "Pad 7", "Pad 8", "Pad 9", "Pad Delete", "Pad Insert", "Pad End", "Pad Down", "Pad Pagedown", "Pad Left", "Pad Clear", "Pad Right", "Pad Home", "Pad Up", "Pad Pageup"

See also

INPUT

Example

This example reads a single keypress from the keyboard and prints the resulting string.

```
dim c as string
do
  c = inkey
loop until c <> ""
print "key pressed: "; c
```

INPUT Statement

Captures a line of data from the keyboard, communications port, or a file, and places the data into a string variable.

Syntax

```
INPUT [#filenum,] variable
```

Remarks

filenum is the file number to use.

variable is the variable name into which you want to place the captured data.

See also

GET, INKEY

Example

This example simply reads a line of input from the user and prints it back out again.

```
dim a as string
input a
print a
```

INSTR Function

Searches for the first occurrence of a string of characters within a specified string.

Syntax

```
INSTR([start,] string1, string2)
```

Remarks

start is the position at which to begin the search. This parameter is optional -- the default is to start searching at position 1.

string1 is the string to search.

string2 is the text to search for within the string.

INSTR returns the position of the first occurrence of the specified text. If INSTR does not find the specified string, it returns a value of 0.

See also

LEFT, MID, RIGHT

Example

This example looks for the string "Windows" within the string "*QmodemPro for Windows*".

```
dim a as string
a = "QmodemPro for Windows"
print instr(a, "Windows")
```

INT Function

Returns the integer or whole number portion of a numeric expression.

Syntax

```
INT(expr)
```

Remarks

FIX and INT operate the same for positive values, however with negative values their operation is slightly different. FIX returns the next-higher integer with negative values, while INT returns the next lower integer with negative values.

See also

FIX

Example

This example shows how the INT function rounds negative numbers down to the next lower integer. In this example the output will be 3 and 5.

```
print int(3.7), int(-4.9)
```

INTEGER Type

Used to declare a variable that can handle integer numbers.

Remarks

Variables of integer type can hold values that range from 32768 to 32767.

See also

BYTE, DIM, LONG, REAL, TYPE

Example

This example declares a variable of type INTEGER and assigns a value to it.

```
dim i as integer
i = 5000
print i
```

KILL Statement

Deletes a file from disk.

Syntax

```
KILL filename
```

Remarks

filename is the name of the file to delete. DOS wildcards * and ? are not supported.

You cannot delete an open file.

This statement is identical to the script DEL command.

See also

<u>DEL</u>, <u>RMDIR</u>

Example

This example removes the file "test.dat" from the current directory.

```
kill "test.dat"
```

LASTCONNECTPASSWORD Function

This function returns the current password.

Syntax

```
LASTCONNECTPASSWORD
```

Remarks

The password returned corresponds to the dialing directory entry that was used for the most recent connection.

See also

DIAL, LASTCONNECTUSERID

Example

```
waitfor "password?"
send lastconnectpassword
```

LASTCONNECTUSERID Function

This function returns the current user id.

Syntax

```
LASTCONNECTUSERID
```

Remarks

The user id returned corresponds to the dialing directory entry that was used for the most recent connection.

See also

DIAL, LASTCONNECTPASSWORD

Example

```
waitfor "name:"
send lastconnectuserid
```

LCASE Function

Returns a copy of a string with all upper case characters converted to lower case.

Syntax

```
LCASE(string)
```

Remarks

string can be any string expression.

This operation is useful for making case insensitive comparisons of text. The UCASE function operates similarly, but converts the specified text to upper case.

See also

UCASE

Example

This example demonstrates using the LCASE function to print a lower case version of a string.

```
dim a as string
a = "QmodemPro for Windows"
print lcase(a)
```

LEFT Function

Returns a string consisting of a specified number of characters starting at the left (beginning) of a string.

Syntax

```
LEFT(string, num)
```

Remarks

string is any string expression.

num is any number in the range 0 through 32767.

If there are fewer than *num* characters in *string*, the entire string will be returned without any padding.

See also

INSTR, MID, RIGHT

Example

This example uses the LEFT function to print the first word in the string "*QmodemPro for Windows*".

```
dim a as string
a = "QmodemPro for Windows"
print left(a, 9)
```

LEN Function

Returns the number of characters in a string, or returns the size in bytes of a user defined type.

Syntax

```
LEN(string)
or
LEN(typename)
```

Remarks

string is any string expression. Used in this way the function will return the number of characters in the string.

typename is the name of any user defined type. Used in this way the function will return the number of bytes taken up by a variable of the indicated type. This is useful when used with the LEN = clause in the OPEN statement.

See also

INSTR, LEFT, MID, OPEN, RIGHT

Example

This example prints the length of the string "*QmodemPro for Windows*". See the OPEN script command for an example of how to use LEN with OPEN.

```
dim a as string
a = "QmodemPro for Windows"
print len(a)
```

LET Statement

Assigns the value of an expression to a variable.

Syntax

```
[LET] var = expr
```

Remarks

The LET keyword is an assignment statement -- it allows you to assign values to variables.

You must declare a variable using DIM before you can assign a value to it.

The keyword itself is optional, so statements such as

```
LET tries = 1
```

is functionally identical to

```
tries = 1
```

The LET keyword is usually omitted in all cases.

See also

DIM, TYPE

Example

This example shows how to declare a variable, assign a value to it, and print its value on the screen.

```
dim i as integer
i = 5
print i
```

LOADPHN Statement

Loads a new phonebook.

Syntax

```
LOADPHN filename
```

Remarks

This command loads a new phone book just like the phonebook's **File/Open** command would.

See also

DIAL

Example

This example shows how to load a new phone book and dial its first entry.

```
loadphn "test.phn"
dial entry 1
```

LOC Function

Returns the current position of a pointer in an open file, indicating the point where the next read or write operation will take place.

Syntax

```
LOC(filenumber)
```

Remarks

filenumber is the number assigned to the open file.

If the file is opened for sequential or binary access, LOC returns the current file position in bytes.

If the file is opened for random access, LOC returns the current record number based on the record size specified in the LEN = clause in the OPEN statement.

See also

GET, OPEN, PUT, SEEK

Example

This example writes an integer to a file and reports the new file position. In this case the file position after writing will be 2.

```
dim i as integer
i = 5
open "test.dat" for random as #1 len = len(integer)
put #1, 1, i
print loc(1)
close #1
```

LOCATE Statement

Positions the cursor on the screen.

Syntax

```
LOCATE row, column
```

Remarks

row is the number of the row at which the cursor is positioned. Rows are numbered 1 through the maximum number of rows on the screen.

column is the number of the column at which the cursor is positioned. Columns are numbered 1 through the number of columns on the screen.

See also

CSRLIN, POS

Example

This example places the cursor on row 5, column 5, and writes a string indicating that fact.

```
locate 5, 5
print "this is at position 5, 5"
```

LOF Function

Returns the number of records in an open file.

Syntax

```
LOF(filenumber)
```

Remarks

filenumber is the number under which the file was opened.

If the file is opened for sequential or binary access, LOF returns the size of the file in bytes.

If the file is opened for random access, LOF returns the size of the file based on the record size specified in the LEN = clause in the OPEN statement.

See also

LOC, OPEN

Example

This example opens a file for sequential access and outputs the length of the file.

```
open "test.dat" for input as #1
print lof(1)
close #1
```

LOG Function

Returns the natural logarithm of a number.

Syntax

```
LOG(expression)
```

Remarks

expression is any numeric expression.

A natural logarithm is the power to which the constant e (about 2.718282) must be raised to obtain a given number. This should not be confused with common logarithms, which are based on 10 rather than e.

If you try to take the logarithm of zero or a negative number, the ERR_MATH error will be generated. You can catch this error with the CATCH statement.

See also

CATCH, EXP

Example

This example prints the logarithm of the number 6.

```
print log(6)
```

LOGFILE Statement

Turns on or off the log file.

Syntax

```
LOGFILE {ON | OFF}
```

Remarks

This command turns on or off the log file just like the **File/Log Toggle** command from the main terminal menu. The log file name is defined in **Options/Files/File Definitions**.

See also

CAPTURE

Example

This example turns on the log file, uploads a file, then turns it off again.

```
logfile on
call upload "c:\qmwin\test.dat", Zmodem
logfile off
```

LONG Type

Used to declare a variable that can handle integer numbers larger than those handled by the INTEGER type.

Remarks

Variables of the long type can hold values that range from 2147483648 to 2147483647.

See also

BYTE, DIM, INTEGER, REAL, TYPE

Example

This example declares a variable of type long, assigns a number to it, and prints out the number.

```
dim i as long
i = 500000
print i
```

LOOP Statement

Used to mark the end of a DO ... LOOP statement. See the DO statement for more information.

LTRIM Function

Trims leading spaces (spaces on the left) from a string expression.

Syntax

```
LTRIM(string)
```

Remarks

string is any string expression.

See also

RTRIM

Example

This example trims the leading spaces off the given string and prints the results within angle brackets.

```
dim a as string
a = "  QmodemPro for Windows  "
print ">"; ltrim(a); "<"
```

MAXIMIZE Statement

Causes *QmodemPro for Windows* to maximize its application window on the Windows desktop.

Syntax

```
MAXIMIZE
```

Remarks

The *QmodemPro for Windows* application window is maximized and brought to the front of the Windows desktop.

See also

<u>ACTIVATE</u>, <u>MINIMIZE</u>, <u>MOVE</u>, <u>SIZE</u>

Example

This example causes the *QmodemPro for Windows* application to maximize its window.

```
MAXIMIZE
```

MID Function

Retrieves a substring from an arbitrary position in another string.

Syntax

```
MID(string, start, num)
```

Remarks

string is any string expression.

start is the position of the first character to copy.

num is the number of characters to return from within the string expression.

If there are not enough characters in the string to supply enough characters in the result, a shorter string than that asked for will be returned.

See also

LEFT, RIGHT

Example

This example prints some characters from the middle of a given string.

```
dim a as string
a = "QmodemPro for Windows"
print mid(a, 7, 11)
```

MINIMIZE Statement

Causes *QmodemPro for Windows* to minimize its application window on the Windows desktop. Minimizing an application makes it into an icon.

Syntax

```
MINIMIZE
```

Remarks

The *QmodemPro for Windows* application window is minimized and its icon is placed near the bottom of the desktop. Any other windows opened by *QmodemPro for Windows* (phonebook, GIF viewer, editor, and so on) will disappear until the application is restored.

See also

<u>ACTIVATE</u>, <u>MAXIMIZE</u>, <u>MOVE</u>, <u>SIZE</u>

Example

This example causes the *QmodemPro for Windows* application to minimize itself and place its icon at the bottom of the screen.

```
MINIMIZE
```

MKDIR Statement

Creates a directory on the disk.

Syntax

```
MKDIR directory
```

Remarks

directory is the name of the directory to create.

When creating a new directory, each of the directories leading up to the last directory name must already exist. If you try to create the directory "c:\abc\def" and the directory "c:\abc" does not already exist, you will get an ERR_PATH error.

See also

CHDIR, CURDIR, RMDIR

Example

This example makes a new directory called "test" under the "c:\qmwin" directory.

```
mkdir "c:\qmwin\test"
```

MOD Operator

Returns the remainder of a division between two integers.

Syntax

```
op1 MOD op2
```

Remarks

op1 and *op2* can be of any integer numeric type (byte, integer, long).

If *op2* is zero the ERR_MATH error will be generated.

Example

This example prints 144, which is the remainder after 400 is divided by 256.

```
print 400 mod 256
```

MOUSECLICK Statement

Used to simulate a mouse click for the RIPscrip emulation.

Syntax

```
MOUSECLICK x, y
```

Remarks

This statement simulates a mouse click on an area of the screen when using the RIPscrip emulation.   The x and y parameters are the horizontal and vertical coordinates of the mouse click.   This command is generated by Quicklearn when using the RIPscrip emulation.

If the RIPscrip emulation is not active, this command has no effect.

See also

EMULATION

Example

This example simulates a mouse click at position 10, 10 on the screen.

```
mouseclick 10, 10
```

MOVE Statement

Used to move the *QmodemPro for Windows* application window on the Windows desktop.

Syntax

```
MOVE x, y
```

Remarks

x is the horizontal coordinate of the new position of the window. X coordinate values range from 0 to the width of your screen minus one.

y is the vertical coordinate of the new position of the window. Y coordinate values range from 0 to the height of your screen minus one.

If you are using standard VGA with a 640x480 screen then the X values can range from 0 through 639 and Y values can range from 0 through 479.

See also

ACTIVATE, MAXIMIZE, MINIMIZE, SIZE

Example

This example moves the *QmodemPro for Windows* application window so its upper left hand corner is at position (300, 200).

```
MOVE 300, 200
```

MSGBOX Statement

Used to pop up a simple message box with a message and an OK button.

Syntax

```
MSGBOX string
```

Remarks

This statement pops up a simple message box with an OK button and waits for the user to press the OK button before continuing.

See also

DIALOG, DIALOGBOX

Example

This example pops up a simple message box on the screen.

```
msgbox "Hello world!"
```

NAME Statement

Renames or moves a file.

Syntax

```
NAME oldfilespec AS newfilespec
```

Remarks

oldfilespec is the current file name.

newfilespec is the new file name.

This command works in a similar way to the DOS RENAME command, with the added ability to move a file, if the new filename includes a path.

You cannot use the NAME command to rename a directory.

If an error occurs during the rename operation, the ERR_FILERENAME error will be generated.

See also

DEL, FINDFIRST, FINDNEXT, KILL

Example

This example renames the file "test.fil" to the file "test.dat".

```
name "test.fil" as "test.dat"
```

NEXT Statement

Marks the end of a FOR ... NEXT loop. See the FOR command for more information and examples.

NOT Operator

Performs a bitwise logical NOT operation.

Syntax

```
NOT op
```

Remarks

This operator takes the binary representation of *op* and reverses the status of each bit in the representation.

When using NOT in an IF statement, the value FALSE (zero) will be converted to TRUE (1) and vice versa.

See also

AND, EQV, IF, IMP, OR, XOR

Example

This example prints out the result of using NOT on various values.

```
print not true
print not false
print not 1
```

OCT Function

Converts an integer expression to an octal (base 8) representation.

Syntax

```
OCT(expr)
```

Remarks

The expression passed to this function will be converted to a long integer before being converted to an octal representation.

See also

HEX, STR, VAL

Example

This example prints the octal representation of the number 42.

```
print oct(42)
```

OPEN Statement

Opens a file so the program can perform input and output operations.

Syntax

```
OPEN file FOR accessmode AS [#]filenum [LEN = reclen]
```

Remarks

file is the name of the file to open.

accessmode specifies the way the program will write to or read from file: INPUT, OUTPUT, APPEND, RANDOM and BINARY.

filenum is the file number to assign to the file.

reclen is the length of a record in a sequential or random-access file, in the range 1 through 32767.

See also

CLOSE, FREEFILE

Example

This example opens a file for sequential output and writes a string of test data to it.

```
open "test.dat" for output as #1
print #1, "this is a test"
close #1
```

OR Operator

OR performs a bitwise logical or operation between its operands.

Syntax

```
op1 OR op2
```

Remarks

op1 and *op2* are logical expressions or numeric values of any integral type (byte, integer, or long integer).

This is the logical truth table for the OR operator:

a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1

See also

AND, EQV, IMP, NOT, XOR

Example

```
print "5 or 6 is 7: "; 5 or 6
```

PAUSE Statement

Used to suspend script execution for a certain time interval.

Syntax

```
PAUSE time
```

Remarks

time is the amount of time to suspend execution, expressed in seconds. If you want to delay for less than a second, use ordinary decimal notation.

This command is identical to the script DELAY command.

See also

DELAY, WAITFOR, WHEN QUIET, WHEN TIME

Example

This example sends "atz" to initialize the modem, pauses for half a second, and sends a command to dial the modem.

```
send "atz"
pause 0.5
send "atdt5551212"
```

PLAY Statement

Used to play a .wav file using the Windows multimedia services.

Syntax

```
PLAY filename
```

Remarks

If Windows cannot find the named .wav file to play, it will sound a default beep (which may be another .wav file). The .wav file is played in "asynchronous" mode which means that Windows will not wait for the sound to finish, and will continue execution with the next script command immediately.

If you would like more control over the options used to play the sound, use the following declaration:

```
declare sub sndPlaySound lib "mmsystem" (fname as string, flags as integer)
```

This declaration declares a new subroutine, sndPlaySound, which actually calls the Windows multimedia sound player directly. For more information on the flags parameter, see the Windows SDK documentation.

See also

BEEP, SOUND

Example

This example plays the sound in the file "soundfil.wav".

```
PLAY "soundfil.wav"
```

POS Function

Reports the current horizontal coordinate position (column number) of the cursor.

Syntax

```
POS
```

Remarks

The return value of this function ranges from 1 to the maximum number of columns on the screen (either 80 or 132).

See also

CSRLIN, LOCATE

Example

This example prints the current horizontal cursor position.

```
print pos
```

PRINT Statement

Outputs data to the display, communications port, or a file.

Syntax

```
PRINT [#filenum, ][expression[[;|,]expression ...]][;|,]
```

Remarks

expression is an expression of any type except a user defined type.

Multiple items can be separated with semicolons (to display the next item immediately after the previous item), or commas (to display the next item in the next 14-character wide "print zone"). A semicolon at the end of the line will suppress the automatic carriage return and line feed that occurs after printing.

The PRINT statement by itself moves the cursor to the beginning of the next line.

See also

LOCATE, TAB

Example

This example shows the famous "Hello world" program in its entirety.

```
print "Hello, world!"
```

PRINTER Statement

Turns the printer toggle on or off.

Syntax

```
PRINTER onoff
```

Remarks

This command turns the printer toggle on and off.   When on, all incoming data is echoed to the printer as well as the screen.   The destination of the printer data is controlled by the "Print file" option in Options/Files/File definitions.

See also

CAPTURE

Example

This example shows how to turn the printer on, wait for some specified data from the port, then turn the printer off again.

```
print on
send
waitfor "Ready"
print off
```

PUT Statement

Writes data to a random access or binary file, from record variables defined by a TYPE ... END TYPE statement.

Syntax

```
PUT [#]filenumber, [position], variable
```

Remarks

filenumber is the file number assigned to the open file.

position is the number of the random access file record, or binary file byte to begin writing. Note that unlike some other languages, the first record or byte in the file is number 1, not number 0. If this position is not specified, then the record is written to the current file position.

variable is the name of the variable that contains the data to place in the file.

See also

GET, INPUT, OPEN, TYPE

Example

This example creates a random access file, writes a record to it from the variable d, and reads it back into the variable z. It then prints out the contents of z to make sure that the operation succeeded.

```
type daterec
  day as integer
  month as integer
  year as integer
end type
dim d as daterec, z as daterec
d.day = 11
d.month = 10
d.year = 1993
open "test.dat" for random as #1 len = len(daterec)
put #1, 1, d
get #1, 1, z
close #1
print z.day, z.month, z.year
```

REAL Type

Used to declare a variable that can handle real (floating-point) numbers.

Remarks

The range of REAL numbers is 3.4e38 to 3.4e38.

See also

BYTE, DIM, INTEGER, LONG, STRING, TYPE

Example

This example declares a variable of type real, assigns a value to it, and prints out the value.

```
dim x as real
x = 1.5
print x
```

RECEIVE Statement

Captures a line of data from the communications port (by default) and places the data into a string variable.

Syntax

```
RECEIVE [#filenum,] var
```

Remarks

This statement is identical to the INPUT statement except that if a file number is not specified, the data is input from the communications port instead of the keyboard.

See also

INPUT

Example

This example reads a line of input from the communications port.

```
dim s as string
receive s
```

RECEIVEFILE Function

Used to receive a file or files from a remote computer.

Syntax

```
RECEIVEFILE(filename, protocol)
```

Remarks

This function initiates a file transfer to receive files from a remote computer. At the time this command is executed, the remote computer must already have started the file transfer. If you are connecting to a bulletin board system (BBS) then your script should already have sent the command to the BBS that will start the file transfer.

protocol is one of the following predefined constants:

ASCII, XMODEM, XMODEMCRC, XMODEM1K, XMODEM1KG, YMODEM, YMODEMG, ZMODEM, KERMIT, BPLUS

For the first five protocols, the *filename* parameter must specify an actual file name in which to place the received file. The ASCII and Xmodem variant protocols do not supply a filename, so one must be supplied in the DOWNLOAD function.

For the last five protocols, the *filename* parameter should be the name of a directory in which to place the received files. If this parameter is an empty string ("") then the download directory specified in **Options/Files/Path Definitions** will be used.

The RECEIVEFILE function returns zero if the file transfer was successful. If the transfer was unsuccessful, RECEIVEFILE returns the error code describing the error. For a list of error codes see Chapter 4.

This function is identical to the DOWNLOAD function.

See also

DOWNLOAD, SENDFILE, UPLOAD

Example

This example receives a file using the Zmodem protocol, assuming the remote computer has already started the transfer.

```
if receivefile("", Zmodem) = 0 then
  print "file transfer ok!"
end if
```

REM Statement

Allows you to include a comment, or non-executing text for your own reference, in a program.

Syntax

```
REM comment
or
' comment
```

Remarks

The apostrophe ' character is a synonym for the REM statement.

Either variation of the statement will cause the compiler to ignore all text on a line following the ' or REM statement.

When adding a comment to a line of executable source code, the REM must be preceded by a colon (:). A colon is not required if you are using the apostrophe.

Comments do not affect the size or execution speed of the compiled script.

Example

This example shows two comment lines and one line that is not a comment.

```
rem This is a comment.
' This is another comment, like the above line.
print "This is not a comment."
```

RESET Statement

Description

Closes all open files.

Syntax

```
RESET
```

Remarks

The RESET statement works the same way as the CLOSE statement without parameters. The functions CLOSE, END, STOP, and SYSTEM also close any open files.

See also

CLOSE, END, STOP, SYSTEM

Example

This example opens a file for input, then closes all open files.

```
open "test.dat" for input as #1
reset
```

RESETEMULATION Statement

Resets the current emulation.

Syntax

```
RESETEMULATION
```

Remarks

This command resets the operation of the currently selected emulation.   It is identical to the Reset Emulation command on the Terminal menu.

See also

EMULATION, RESETMODEM

Example

This example shows how the RESETEMULATION command might be used.

```
hangup
resetemulation
```

RESETMODEM Statement

Sends the modem reset string as specified in the modem configuration.

Syntax

```
RESETMODEM
```

Remarks

This command sends the "Modem init" string as specified in the modem configuration.

See also

AUTOANSWER, RESETEMULATION

Example

This example shows how the RESETMODEM command might be used to reset the modem after hanging up.

```
hangup
resetmodem
```

RETURN Statement

Returns from a subroutine started with the GOSUB statement. See the GOSUB statement for more information and examples.

RIGHT Function

Returns a string consisting of a specified number of characters starting at the right (end) of a string.

Syntax

```
RIGHT(string, num)
```

Remarks

string is any string expression.

num is any number in the range 0 through 32767.

If there are fewer than num characters in the string, the entire string is returned even though it is shorter than num.

See also

INSTR, LEFT, MID

Example

This example shows how the RIGHT function can be used to extract the rightmost characters from another string.

```
dim a as string
a = "QmodemPro for Windows"
print right(a, 7)
```

RMDIR Statement

Removes a directory from a disk.

Syntax

```
RMDIR directory
```

Remarks

directory is the name of the directory to remove.

You cannot remove a directory if it still contains files.

If the directory cannot be removed, the ERR_PATH error is generated. Use the CATCH statement to respond to this error.

See also

CHDIR, KILL, MKDIR

Example

This example removes the directory "c:\qmwin\test", assuming it is empty.

```
rmdir "c:\qmwin\test"
```

RND Function

Returns a pseudo-random real number between greater than or equal to 0 and less than 1.

Syntax

```
RND[(number)]
```

Remarks

If *number* is omitted or greater than zero, the next random number in sequence is generated.

If *number* is less than zero, RND returns the same random number every time, depending on the value of expr.

If *number* is zero, the last random number generated is returned.

Example

This example shows the three ways of using the RND function.

```
print rnd(-0.5)
print rnd
print rnd(0)
```

RTRIM Function

Trims trailing spaces (spaces on the right) from a string expression.

Syntax

```
RTRIM(string)
```

Remarks

string is any string expression. Any trailing spaces are removed from the string and the result is returned.

See also

LTRIM

Example

This example shows how the RTRIM function can be used to trim trailing spaces from a string.

```
dim a as string
a = "  QmodemPro for Windows  "
print ">"; rtrim(a); "<"
```

SCREEN Function

Returns the character or color attribute at a particular screen position.

Syntax

```
SCREEN(x, y [, attr])
```

Remarks

Returns the character or attribute (color) at a particular screen position. The screen position is given by the *x* and *y* parameters and ranges from 1 to the with or height of the terminal screen.

If the *attr* parameter is omitted or is zero, this function returns the ASCII value of the character on the screen at the given position. If the attr parameter is one, the return value is the attribute of the character on the screen at the given position.

The attribute returned is defined by the following formula:

```
b * 16 + f
```

where b is the background color and f is the foreground color, as listed under the COLOR statement.

See also

COLOR

Example

This example clears the screen, writes "Hi" in gray on blue, and prints the ASCII value of "H" followed by the color value 71.

```
cls
color 7, 4
print "Hi"
print screen(1, 1), screen(2, 1, 1)
```

SCROLLBACK Statement

Turns scrollback mode on or off.

Syntax

```
SCROLLBACK onoff
```

Remarks

This command turns the scrollback mode on or off.   The operation of the script is not affected while in scrollback mode (ie. the script continues to execute).

See also

CAPTURE, SCROLLBACKRECORD

Example

This example shows how to capture some data, then automatically go to scrollback mode to review it.

```
send
waitfor "Command"
scrollback on
```

SCROLLBACKRECORD Function

Used to get or set the current state of the scrollback record toggle.

Syntax

```
SCROLLBACKRECORD
or
SCROLLBACKRECORD(onoff)
```

Remarks

There are two forms to the SCROLLBACKRECORD function. The first form takes no arguments and simply returns the current setting. The second form sets the state of the toggle and returns the previous state.

In both the parameter and the return value, a nonzero value means scrollback record mode is turned on.   A zero value means scrollback record mode is turned off.

When scrollback record mode is on, incoming data is captured to the scrollback buffer for later review.   When scrollback record mode is turned off, data is not copied to the scrollback buffer.

See also

CAPTURE, SCROLLBACK

Example

This example shows how the SCROLLBACKRECORD function might be used to automatically capture some data.

```
scrollbackrecord on
send
waitfor "End of data"
scrollbackrecord off
```

SEEK Statement

Sets the current position in an open file for the next I/O operation.

Syntax

```
SEEK [#]filenumber, position
```

Remarks

filenumber is the number allocated to the open file.

position is the number of the record or byte to be accessed. Note that unlike many programming languages, the records and bytes in a file are numbered starting at 1 rather than at 0.

See also

GET, LOC, OPEN, PUT

Example

This example reads from "c:\autoexec.bat" the string starting at offset 50.

```
dim s as string
open "c:\autoexec.bat" for input as #1
seek #1, 50
input #1, s
close #1
```

SELECT CASE Statement

Allows a number of expressions to be compared, executing statements based on the results of the comparison.

Syntax

```
SELECT CASE expression
  CASE case-condition[, case-condition ...]
    [statements]
  [CASE case-condition[, case-condition ...]
    [statements]]
  [CASE ELSE
    [statements]]
END SELECT
```

Remarks

expression is a numeric or string expression.

case-condition is a condition in one of the following forms:

```
expression
expression TO expression
IS relational-operator expression
```

If the *expression* matches the *case-condition*, the statements matching the corresponding CASE are executed. Control then passes to the first statement following END SELECT.

If none of the *case-conditions* matches and there is a CASE ELSE clause, the statements within the CASE ELSE clause are executed.

See also

IF

Example

This example tests the numbers 0 through 11 with a SELECT CASE statement and prints the results.

```
dim i as integer
for i = 0 to 11
  select case i
    case 1
      print "i is 1"
    case 2 to 9
      print "i is between 2 and 9 inclusive"
    case is >= 10
      print "i is greater than or equal to 10"
    case else
```

```
        print "i must be zero or negative"
    end select
next i
```

SEND Statement

Outputs data to the communications port.

Syntax

```
SEND [expression[[;|,]expression ...]][;|,]
```

Remarks

expression is an expression of any type except a user defined type.

Multiple items can be separated with semicolons (to display the next item immediately after the previous item), or commas (to display the next item in the next 14-character wide print zone).

A semicolon at the end of the line will suppress the automatic carriage return after the transmitted data.

The SEND statement by itself sends only a carriage return.

See also

LASTCONNECTPASSWORD, LASTCONNECTUSERID, PRINT

Example

This example shows how the SEND command may be used to send various data to the modem.

```
send "bob"        ' sends "bob" followed by Enter
send "g";         ' sends "g" without an Enter
send lastconnectuserid
                  ' sends the UserId corresponding
                  ' to the phone directory entry
                  ' you are currently connected to
```

SENDFILE Function

Used to send files to a remote computer.

Syntax

```
SENDFILE(filename, protocol)
```

Remarks

This function initiates a file transfer to send files to a remote computer. At the time this command is executed, the remote computer must already have started the file transfer. If you are connecting to a bulletin board system (BBS) then your script should already have sent the command to the BBS that will start the file transfer.

The protocol is one of the following predefined constants:

ASCII, XMODEM, XMODEMCRC, XMODEM1K, XMODEM1KG, YMODEM, YMODEMG, ZMODEM, KERMIT, BPLUS

The first five protocols require a single file name in the filename parameter. This file name indicates the file to send.

The last five protocols can accept more than just one file in the filename parameter -- separate multiple filenames with spaces.

The SENDFILE function returns zero if the file transfer was successful. If the transfer was unsuccessful, SENDFILE returns the error code describing the error. For a list of error codes see Chapter 4.

This function is identical to the UPLOAD function.

See also

DOWNLOAD, RECEIVEFILE, UPLOAD

Example

This example shows how to send a file using the Zmodem protocol, assuming the remote computer has been told to accept a file transfer.

```
if sendfile("c:\qmwin\test.dat", Zmodem) = 0 then
  print "file transfer ok!"
end if
```

SETCOMM Statement

Sets the communications parameters without changing the active device.

Syntax

```
SETCOMM commsettings
```

Remarks

The *commsettings* parameter should be a string in the following format:

```
"baudrate,DPS"
```

baudrate is the baud rate.

D is the number of data bits.

P is the type of parity: N for None, E for Even, O for Odd, M for Mark, or S for Space.

S is the number of stop bits.

See also

DEVICE

Example

This example sets the communications port to 19200 baud, 8 data bits, no parity bit, and one stop bit.

```
setcomm "19200,8N1"
```

SETDTR Statement

Sets the DTR (Data Terminal Ready) signal high (ON) or low (OFF).

Syntax

```
SETDTR on | off
```

Remarks

This command is provided to allow direct control over the DTR (Data Terminal Ready) line on the modem for special applications.   Normally lowering DTR (SETDTR OFF) will hang up the modem   the same action as the HANGUP command with the modem hangup string set to "DTR".

See also

HANGUP

Example

This example sends a command to log off an on-line service, waits five seconds, then hangs up the modem.

```
send "bye"
delay 5
setdtr off
```

SGN Function

Returns an integer indicating the sign of a number.

Syntax

```
SGN(expr)
```

Remarks

If expr is greater than zero, the return value is 1. If expr is less than zero, the return value is 1. If expr is equal to zero, the return value is zero.

See also

ABS

Example

This example shows what happens to a negative number, zero, and a positive number with the SGN function.

```
print sgn(-5), sgn(0), sgn(3)
```

SHELL Statement

Runs a DOS command or other executable program or batch file from a script.

Syntax

```
SHELL [command]
```

Remarks

command is a string expression containing an executable DOS or Windows command.

See also

END, SYSTEM

Example

This example runs Windows Control Panel.

```
shell "control"
```

SIN Function

Returns the sine of an angle.

Syntax

```
SIN(angle)
```

Remarks

angle is the measurement of an angle expressed in radians. You can compute radians to degrees by multiplying by 180/pi (pi is approximately 3.14159).

See also

ATN, COS, TAN

Example

This example prints the sine of 1 radian.

```
print sin(1)
```

SIZE Statement

Used to resize the *QmodemPro for Windows* application window on the Windows desktop.

Syntax

```
SIZE width, height
```

Remarks

width is the new width of the window on the desktop. Width values range from 0 to the width of your screen.

Height is the new height of the window on the desktop. Height values range from 0 to the height of your screen.

If you are using standard VGA with a 640x480 screen then the width values can range from 0 through 640 and height values can range from 0 through 480.

See also

ACTIVATE, MAXIMIZE, MINIMIZE, MOVE

Example

This example changes the size of the *QmodemPro for Windows* application window to be 400 pixels wide and 300 pixels high.

```
SIZE 400, 300
```

SOUND Statement

Generates a tone of a specific frequency and duration.

Syntax

```
SOUND frequency, duration
```

Remarks

This statement generates a tone of a specific frequency and duration. The frequency is specified in hertz (cycles per second) and the duration is specified in seconds.

See also

BEEP, PLAY

Example

This example sounds a 1000 Hz tone for 2/10 of a second.

```
SOUND 1000, 0.2
```

SPACE Function

Returns a string consisting of a specified number of spaces.

Syntax

```
SPACE(number)
```

Remarks

number is the number of spaces to return.

This function is identical to the SPC function.

See also

SPC, TAB

Example

This example prints two words separated by 20 spaces.

```
print "here"; space(20); "there"
```

SPC Function

Returns a string consisting of a specified number of spaces.

Syntax

```
SPC(number)
```

Remarks

number is the number of spaces to return.

This function is identical to the SPACE function.

See also

SPACE, TAB

Example

This example prints two phrases separated by 30 spaces.

```
print "neither here"; spc(30); "nor there"
```

SPLITSCREEN Statement

Turns split screen mode on or off.

Syntax

```
SPLITSCREEN onoff
```

Remarks

This command turns the split screen mode on or off.   The operation of the script is not affected while in split screen mode (ie. the script continues to execute).

See also

SCROLLBACK

Example

This example shows how to turn on and off the split screen mode from a script.

```
splitscreen on
splitscreen off
```

SQR Function

Returns the square root of a numeric expression.

Syntax

```
SQR(expression)
```

Remarks

expression must evaluate to a number greater than or equal to zero.

If the expression is less than zero, SQR will generate the ERR_MATH error. This error can be caught with the CATCH statement.

See also

EXP, LOG

Example

This example prints the square root of 9.

```
print sqr(9)
```

STAMP Statement

Used to write a string of text to the log file, if it is open.

Syntax

```
STAMP string
```

Remarks

string is a string that is written to the log file. If the log file is not currently open, this statement has no effect.

See also

CAPTURE, LOGFILE

Example

This example opens the log file, writes a stamp string to it, and closes the log file.

```
logfile on
stamp "This is a stamped string."
logfile off
```

STATIC Statement

Used to declare variables in a subroutine or function that retain their values even after the subroutine or function has returned to its caller. The variables can be used when the subroutine or function is called again without being reinitialized to zero or a blank string.

Syntax

```
STATIC var[([lowerbound TO] upperbound)] [AS type][, var[([lowerbound
TO] upperbound)] [AS type]...]
```

Remarks

The syntax of the STATIC statement is identical to that of the DIM statement. For a complete discussion of variable declarations please see the DIM statement.

See also

DIM

Example

This example shows how a STATIC variable might be used to count the number of times a function has been called.

```
declare sub test
call test
call test

sub test
  static i as integer
  i = i + 1
  print "test has been called "; i; " times."
end sub
```

STOP Statement

Terminates execution of a script.

Syntax

```
STOP
```

Remarks

The STOP statement is similar to the END statement except that it puts a warning message on the screen after the script has terminated.

See also

END

Example

This example uses STOP to terminate a loop.

```
dim i as integer
i = 0
do
  i = i + 1
  if i > 99 then stop
loop
```

STR Function

Converts a number to a string representation of the number.

Syntax

```
STR(number)
```

Remarks

This function returns the string representation of *number*.

This function is the opposite of the VAL function, which converts a number in string format to an integer.

See also

VAL

Example

This example uses the STR function to convert a number to a string and count the number of digits in the number.

```
dim a as string
a = str(47)
print a; "is "; len(a); " digits long."
```

STRING Function

Repeats a character a specified number of times, and returns the results as a string.

Syntax

```
STRING(number, code)
or
STRING(number, string)
```

Remarks

number is the number of characters to output, in the range 0 to 32767.

code is the ASCII code for the character you want, in the range 0 to 255.

In the second syntax, *string* is a string and the STRING function returns the first character of *string* repeated *num* times.

See also

CHR, SPACE

Example

This example prints a string of 50 dashes on the screen.

```
print string(50, "-")
```

STRING Type

Used to declare a variable that can handle variable length character data, or to define a string that can hold a specific number of characters.

Remarks

Variables of string type can hold values that are up to 32767 characters in length.

To declare a fixed length string that holds only a specific number of characters, use the syntax

```
STRING*n
```

where n is the number of characters you want the string to hold. Fixed length strings are useful because variable length strings are not allowed within user defined type declarations.

See also

DIM, TYPE

Example

This example declares a string, assigns a value to it, and prints the value.

```
dim a as string
a = "Hello, world!"
print a
```

STRIPHIBIT Function

Used to get or set the current state of the "8th Bit Strip" toggle.

Syntax

```
STRIPHIBIT
or
STRIPHIBIT(onoff)
```

Remarks

There are two forms to the STRIPHIBIT function. The first form takes no arguments and simply returns the current setting. The second form sets the state of the toggle and returns the previous state.

In both the parameter and the return value, a nonzero value means the high bit is stripped from incoming characters. A zero value means the high bit is not stripped.

See also

DUPLEX

Example

This example shows how the STRIPHIBIT function can be used in a logon script to turn on the high bit strip.

```
dim oldstrip as integer
oldstrip = striphibit(on)
waitfor "UserID"
send "123456"
striphibit oldstrip
```

SUB Statement

Allows you to define your own subprograms that do not directly return a value to the caller.

Syntax

```
SUB name[(arg AS type[, arg AS type]...)]
   ...
END SUB
```

Remarks

name is the name you assign to the subroutine.

arg is the name of a formal argument to the subroutine. Each argument must have an associated type declaration using the AS keyword.

You may define local variables within your user-defined subroutine. You can use STATIC declarations to preserve the value of individual variables across subroutine calls.

You can execute a sub-program using the CALL statement, along with any arguments you wish to pass to the sub-program.

See also

CALL, DECLARE, EXIT, FUNCTION, STATIC

Example

This example defines a simple subroutine and shows two ways of calling it.

```
sub test(i as integer)
  print "in subroutine test: "; i
end sub

sub 1
call sub(5)
```

SYSTEM Statement

Closes all open files and terminates a script.

Syntax

```
SYSTEM
```

Remarks

This statement immediately ends the script and closes the *QmodemPro for Windows* application.

See also

END, SHELL, STOP

Example

This example terminates *QmodemPro for Windows* after waiting for a key to be pressed.

```
print "press a key to exit...";
while inkey = ""
wend
system
```

TAB Function

Returns the appropriate number of spaces to move the cursor to the specified column on the screen.

Syntax

```
TAB(column)
```

Remarks

This function returns the appropriate number of spaces to move the cursor to a particular column on the screen. Note that this function only works on the actual terminal screen.

See also

SPC

Example

This example shows how the TAB function might be used to print columnar data on the screen.

```
print "test"; tab(40); "column 40"
```

TAN Function

Returns the tangent of an angle.

Syntax

```
TAN(angle)
```

Remarks

angle is the measurement of an angle expressed in radians. You can compute radians to degrees by multiplying by 180/pi (pi is approximately 3.14159).

See also

<u>ATN</u>, <u>COS</u>, <u>SIN</u>

Example

This example prints the tangent of 1 radian.

```
print tan(1)
```

TIME Function

Returns the current date from the computer's internal clock.

Syntax

```
TIME
```

Remarks

The TIME function returns the current time in the form specified in Windows Control Panel, International section.

See also

DATE

Example

This example prints the current time according to Windows.

```
print "The time is now "; time
```

TIMEOUT Function

Used to either set or retrieve the current script timeout setting.

Syntax

```
TIMEOUT
or
TIMEOUT(seconds)
```

Remarks

When TIMEOUT is called without an argument, it returns the current script timeout value.

When TIMEOUT is called with a *seconds* argument, it sets the current timeout value to the specified number of seconds. In addition, it returns the previous timeout value.

If the timeout is set to zero, the timeout function is disabled.

The script timeout value applies to INPUT, RECEIVE, and WAITFOR commands.

See also

INPUT, RECEIVE, WAITFOR, WHEN

Example

This example shows how you might modify the timeout value during a logon script.

```
timeout 20
waitfor "what is your first name"
send "joe"
...
catch err_timeout
hangup
end
```

TIMER Function

Returns the number of seconds since midnight.

Syntax

```
TIMER
```

Remarks

TIMER returns the number of seconds since midnight as a real type value.

See also

TIMEOUT, WAITFOR, WHEN TIME

Example

This example prints the number of seconds since midnight, according to Windows.

```
print timer
```

TRAPSCREEN Statement

Used to capture the current terminal screen to a file.

Syntax

```
TRAPSCREEN filename
```

Remarks

This function takes a snapshot of the current terminal screen and appends it to the named file.   A date and time stamp precedes the data on the screen.

See also

CAPTURE, SCROLLBACK

Example

This example shows how the TRAPSCREEN command might be used to capture some data from the remote host.

```
send
waitfor "OK"
trapscreen "snapshot.txt"
```

TYPE ... END TYPE Statement

Allows you to create a data structure (data type or user-defined type) consisting of one or more fields.

Syntax

```
TYPE typename
  field AS typename
  ...
END TYPE
```

Remarks

typename is the name of the new type.

fieldname is an element of the user-defined data type.

typename is the type of the field. It can be of type BYTE, INTEGER, LONG, REAL, or STRING*n (variable length strings are not permitted in user-defined types), or another user-defined type.

Variables of your new user-defined type are declared just like regular variables using the DIM statement. To access a field of your user defined variable, use a period after the variable name followed by the field name. See the example for an illustration of this.

See also

DIALOG, DIM

Example

This example declares a user-defined type called "daterec", declares a variable d of the type, assigns values to each of its fields, and prints the values of each of its fields.

```
type daterec
  day as integer
  month as integer
  year as integer
end type
dim d as daterec
d.day = 11
d.month = 10
d.year = 1993
print d.day, d.month, d.year
```

UCASE Function

Returns a copy of a string with all lower case characters converted to upper case.

Syntax

```
UCASE(string)
```

Remarks

string can be any string expression.

This operation is useful for making case insensitive comparisons of text. The LCASE function operates similarly, but converts the specified text to lower case.

See also

LCASE

Example

This example shows how the UCASE function converts all lower case characters in a string to upper case.

```
dim a as string
a = "QmodemPro for Windows"
print ucase(a)
```

UPLOAD Function

Used to send files to a remote computer.

Syntax

```
UPLOAD(filename, protocol)
```

Remarks

This function initiates a file transfer to send files to a remote computer. At the time this command is executed, the remote computer must already have started the file transfer. If you are connecting to a bulletin board system (BBS) then your script should already have sent the command to the BBS that will start the file transfer.

protocol is one of the following predefined constants:

ASCII, XMODEM, XMODEMCRC, XMODEM1K, XMODEM1KG, YMODEM, YMODEMG, ZMODEM, KERMIT, BPLUS

The first five protocols require a single file name in the *filename* parameter. This file name indicates the file to send.

The last five protocols can accept more than just one file in the *filename* parameter -- separate multiple filenames with spaces.

The UPLOAD function returns zero if the file transfer was successful. If the transfer was unsuccessful, UPLOAD returns the error code describing the error. For a list of error codes see Chapter 4.

This function is identical to the SENDFILE function.

See also

DOWNLOAD, RECEIVEFILE, SENDFILE

Example

This example starts a file transfer of "c:\qmwin\test.dat" to a remote computer. It assumes that the remote computer is prepared to receive the file.

```
if upload("c:\qmwin\test.dat", Zmodem) = 0 then
  print "file transfer ok!"
end if
```

VAL Function

Returns the numeric value of a string.

Syntax

```
VAL(string)
```

Remarks

The VAL function begins evaluating a string at the first character, and scans it until the end of the string, or until a non-numeric character is reached, whichever comes first. It returns the value of the number found in the string, if any (if no number is found, zero is returned).

The VAL function does not support REAL numbers.

See also

STR

Example

This example converts the string containing the digits 4 and 2 to the decimal value 42.

```
dim a as string, i as integer
a = "42"
i = val(a)
print i
```

VERSION Function

Returns the *QmodemPro for Windows* version number as a string.

Syntax

```
VERSION
```

Remarks

In version 1.00 of *QmodemPro for Windows*, this function returns the string "1.00". You can always assume that the version numbers will be increasing with respect to the string comparison operators.

See also

<u>DOSVERSION</u>, <u>WINVERSION</u>

Example

This example prints the current *QmodemPro for Window* version number.

```
print "QmodemPro for Windows "; version
```

VIEWFILE Statement

Used to invoke the internal file viewer with a specified file.

Syntax

```
VIEWFILE filename
```

Remarks

This function brings up the internal file viewer with the named file loaded ready for viewing.   Note that that script continues to run after the viewer is started.

See also

EDITFILE, VIEWGIF

Example

This example shows how the VIEWFILE command might be used to view the host mode user file.

```
viewfile "host.usr"
```

VIEWGIF Statement

Used to invoke the internal GIF viewer with a specified file.

Syntax

```
VIEWGIF filename
```

Remarks

This function brings up the internal GIF viewer with the named GIF or BMP file loaded ready for viewing.   Note that that script continues to run after the viewer is started.

See also

EDITFILE, VIEWFILE

Example

This example shows how the VIEWGIF command might be used to view a GIF picture.

```
viewgif "fastcar.gif"
```

WAITFOR Statement

Used to wait for a particular sequence of characters to appear from the communications port.

Syntax

```
WAITFOR string [, timeout]
```

Remarks

WAITFOR suspends script execution until the specified string appears from the communications port. By default, the timeout value is whatever was last set by the TIMEOUT command. If a timeout value is specified, it is used instead of the current TIMEOUT value.

The case of letters is ignored when comparing incoming data against the specified string. However, terminal emulation escape sequences are not ignored.

If the character string does not appear within the timeout period, the ERR_TIMEOUT error is generated. This can be caught with the CATCH script statement to take corrective action.

See also

RECEIVE, WHEN

Example

This example shows how the WAITFOR command might be used in a script to log on to an on-line service.

```
waitfor "first name"
send "john doe"
waitfor "password"
send "sesame"
```

WEND Statement

Used to mark the end of a WHILE ... WEND statement. See the WHILE statement for more information and examples.

WHEN CLEAR Statement

Used to remove all when conditions, a set of when conditions, or a particular when condition.

Syntax

```
WHEN CLEAR ALL
or
WHEN CLEAR name
```

Remarks

WHEN CLEAR ALL clears all the currently active WHEN conditions.

WHEN CLEAR name clears all the WHEN conditions that have the name *name*.

After a WHEN condition is cleared, it is no longer active.

See also

WHEN DISABLE, WHEN ENABLE, WHEN

Example

This example shows how a WHEN CLEAR command might be used to remove a WHEN MATCH condition that is no longer needed.

```
when match "press enter" do send
waitfor "main menu"
when clear all
```

WHEN DISABLE Statement

Used to disable all when conditions, a set of when conditions, or a particular when condition.

Syntax

```
WHEN DISABLE ALL
or
WHEN DISABLE name
```

Remarks

WHEN DISABLE ALL disables all the currently active WHEN conditions.

WHEN DISABLE name disables all the WHEN conditions that have the name *name*.

A disabled WHEN condition can be reactivated with the WHEN ENABLE statement.

See also

WHEN CLEAR, WHEN ENABLE, WHEN

Example

This example shows how the when condition with the name "Enter" might be disabled if it is temporarily unnecessary.

```
when name "Enter" match "press enter" do send

...
when disable "Enter"
```

WHEN ENABLE Statement

Used to enable all when conditions, a set of when conditions, or a particular when condition.

Syntax

```
WHEN ENABLE ALL
or
WHEN ENABLE name
```

Remarks

WHEN ENABLE ALL enables all the currently active WHEN conditions.

WHEN ENABLE name enables all the WHEN conditions that have the name *name*.

See also

WHEN CLEAR, WHEN DISABLE, WHEN ENABLE, WHEN

Example

This example shows how the WHEN ENABLE command might be used to turn on a WHEN MATCH event that has been previously disabled.

```
when name "Enter" match "press enter" do send
...
when disable "Enter"
...
when enable "Enter"
```

WHEN Statement

Used set up a mechanism that will trigger a particular statement or set of statements when a certain event happens.

Syntax

```
WHEN [NAME name] {MATCH string | QUIET seconds | TIME time} DO
statements
or
WHEN [NAME name] {MATCH string | QUIET seconds | TIME time} DO
  statements ...
END WHEN
```

Remarks

There are three kinds of WHEN trigger events: MATCH, QUIET, and TIME.

A MATCH event watches the communications port for a particular sequence of characters and executes the statements associated with the WHEN statement when the sequence of characters is seen. Upper and lower case characters are considered the same when comparing characters.

A QUIET event watches the activity on the communications port and executes the statements associated with the WHEN statement when there is no activity for a certain amount of time. This is often used to send an Enter whenever there is no data received from the remote computer for, say, 10 seconds.

A TIME event watches the time and executes the statements associated with the WHEN statement when a particular time of day occurs. The time of day is specified in string format, so "2:00" would be 2:00am. Hours, minutes, and seconds are accepted separated by colons. An AM or PM indicator is also accepted.

See also

WAITFOR, WHEN CLEAR, WHEN DISABLE, WHEN ENABLE

Example

This example shows how the WHEN MATCH command might be used to skip past logon screens when waiting for a "main menu" prompt from an on-line service.

```
when match "press enter" do send
waitfor "main menu"
```

WHILE ... WEND Statement

Repeats a series of statements while a given condition is true.

Syntax

```
WHILE expression
   [statements]
WEND
```

Remarks

expression is any numeric expression or variable that evaluates to a nonzero value (true) or zero (false). A WHILE ... WEND loop will execute so long as the expression is true, then when the expression becomes false, then passes control to the statement that follows the WEND.

The WHILE ... WEND provides an alternate syntax to the DO WHILE ... LOOP statement.

Each WHILE must have a corresponding WEND. Unmatched statements will cause an error message.

See also

DO ... LOOP, FOR ... NEXT

Example

This example shows how a WHILE loop might be used to print out the first 10 whole numbers.

```
dim i as integer
i = 0
while i < 10
  print "i is "; i
  i = i + 1
wend
```

WINVERSION Function

Returns the current Windows version as a string.

Syntax

```
WINVERSION
```

Remarks

This function returns the Windows version as a string in the format X.Y. X is the major version number and Y is the minor revision number.

For example, Windows version 3.1 would be reported as "3.10".

See also

DOSVERSION, VERSION

Example

This example prints out the current Windows version number.

```
print "Currently using Windows "; winversion
```

XONXOFF Statement

Used to change the state of the Xon/Xoff (software flow control) setting.

Syntax

```
XONXOFF onoff
```

Remarks

This function turns on or off Xon/Xoff (software flow control) from a script.

See also

DEVICE

Example

This example shows how the XONXOFF command might be used to turn on software flow control in a logon script.

```
waitfor "UserID"
send "123456"
xonxoff on
```

XOR Operator

XOR performs a bitwise logical exclusive-or operation between its operands.

Syntax

```
op1 XOR op2
```

Remarks

op1 and op2 are logical expressions or numeric values of any integral type (byte, integer, or long integer).

This is the logical truth table for the XOR operator:

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

See also

AND, EQV, IMP, NOT, OR

Example

This example shows how the XOR operator is used to do a bitwise exclusive-or operation.

```
print "5 xor 6 is 3: "; 5 xor 6
```

Using Dialog Boxes

The *QmodemPro for Windows* script language allows you to create Windows dialog boxes that you can use to gather information from the user, display status information, and so on.

There are two steps that are necessary to create dialog boxes using the *QmodemPro for Windows* script language:

Create the dialog box template.

Create a dialog box variable based on the template.

Creating a Dialog Box Template

A dialog box template declaration is very similar to a user defined type declaration. Here is the syntax for a dialog type declaration:

```
DIALOG dialogtype x, y, w, h
  [CAPTION caption]
  [FONT size, fontname]
  [integer-field AS CHECKBOX title, id, x, y, w, h]
  [integer-field AS COMBOBOX id, x, y, w, h]
  [CTEXT title, id, x, y, w, h]
  [DEFPUSHBUTTON title, id, x, y, w, h]
  [string-field as EDITTEXT id, x, y, w, h]
  [GROUPBOX title, id, x, y, w, h]
  [integer-field AS LISTBOX id, x, y, w, h]
  [LTEXT title, id, x, y, w, h]
  [PUSHBUTTON title, id, x, y, w, h]
  [integer-field AS RADIOBUTTON title, id, x, y, w, h]
  [RTEXT title, id, x, y, w, h]
  ...
END DIALOG
```

The DIALOG keyword is used to introduce a dialog box declaration. Following this keyword is the name of the dialog type. After the name are four numbers which define the position and size of the dialog:

> x -- the horizontal position of the dialog relative to the top left corner of the screen

> y -- the vertical position of the dialog relative to the top left corner of the screen

> w -- the width of the dialog in "dialog units"

> h -- the height of the dialog in "dialog units"

Notice that the width and height of the dialog are specified in "dialog units" -- these are units that are based on the size of the font used in the dialog. A horizontal dialog unit is approximately 1/4 of the average width of the characters in the font, and a vertical dialog unit is 1/8 of the height of the characters in the font. Dialog units are specified in this way to avoid scaling problems when changing the font used for the dialog box.

Most dialog boxes will include a CAPTION statement that includes the title of the dialog box. If no CAPTION statement is included, the title will be blank. For example,

```
caption "My dialog box"
```

The FONT statement allows you to change the font used to display the dialog box. Any point size and face name may be specified here -- the actual font used depends on the fonts installed in your Windows system (some types of fonts, like the Script font, available elsewhere in Windows are not available for use in dialog boxes). For example, to use 10 point Arial:

```
font 10, "Arial"
```

The remainder of the commands allowed in dialog box definitions define what are called "controls". Controls have some common features -- each control must have a unique positive ID number or 1 if an ID number is not necessary, and must have a position and size defined by the x, y, w, h parameters. Specific features and requirements of each of the control types are discussed in the following paragraphs.

CHECKBOX

Defines a check box control that can be set either on or off. The title parameter defines the caption that appears to the right hand side of the check box itself. A positive ID number is required for this control type. The integer field associated with this control is 1 if the check box is checked, otherwise it is zero.

COMBOBOX

Defines a combo box control that offers a list of choices. A positive ID number is required for this control type. The integer field associated with this control indicates the number of the choice that was picked. To fill the combo box with choices, a dialog INIT procedure must be defined (see below).

CTEXT

Defines a centered-text control that displays text in the dialog box. The title defines the text to display. An ID number is not required for this control so it should be set to 1.

DEFPUSHBUTTON

Defines the "default" push button in the dialog (the default push button is the one that gets pressed when the user presses Enter). The title defines the text that will appear on the face of the button. A positive ID number is required for this control type. There should only be one DEFPUSHBUTTON in the dialog box template.

EDITTEXT

Defines a text edit control that allows the user to enter or edit text. A positive ID number is required for this control type. The string field associated with this control is the actual text that appears within the edit control.

GROUPBOX

Defines a shaded box control that can be used to group controls together visually. An ID number is not required for this control so it should be set to 1.

LISTBOX

Defines a list box control that offers a list of choices. A positive ID number is required for this control type. The integer field associated with this control indicates the number of the

choice that was picked. To fill the list box with choices, a dialog INIT procedure must be defined (see below).

LTEXT

Defines a left-justified text control that displays text in the dialog box. The title defines the text to display. An ID number is not required for this control so it should be set to 1.

PUSHBUTTON

Defines a push button in the dialog. The title defines the text that will appear on the face of the button. A positive ID number is required for this control type. If the ID number is not IDOK or IDCANCEL, then a response function should be defined for the push button so that some action can be taken when the button is pressed (IDOK and IDCANCEL automatically cause the dialog box to close and return IDOK or IDCANCEL to the script).

RADIOBUTTON

Defines a radio button control that can be set either on or off, with the additional property that only one radio button in a dialog may be "on" at any time. The title parameter defines the caption that appears to the right hand side of the radio button itself. A positive ID number is required for this control type. The integer field associated with this control is 1 if the radio button is checked, otherwise it is zero.

RTEXT

Defines a right-justified text control that displays text in the dialog box. The title defines the text to display. An ID number is not required for this control so it should be set to 1.

After each of the desired controls is defined, the dialog box template is ended with the END DIALOG keywords.

A Word About Control IDs

The positive ID numbers used by many of the above control types are used internally by Windows to manage the dialog box once it is displayed on the screen. For this reason, certain rules should be followed to ensure that the dialog box is displayed properly.

Some ID numbers with low values are reserved by Windows and should not be used except where they are intended to be used. To avoid problems, a good place to start numbering your controls such as check boxes and edit controls is at number 101 and continuing upwards. ID numbers should never be used more than once in the same dialog box.

Buttons generally follow the same rules as other controls except for the special ID values IDOK (equal to 1) and IDCANCEL (equal to 2). When the IDOK or IDCANCEL values are used with a button, that button will automatically cause the dialog box to close and return the corresponding ID number. Many dialog boxes just have two buttons, Ok and Cancel, and these ID values provide a quick way to make those buttons work as expected. Buttons defined with other ID values will not automatically cause the dialog box to close - to perform some action requires a dialog box response function (below).

Using Dialog Box Variables

Once a dialog box template has been constructed and defined, it is easy to make a dialog box appear on the screen. Suppose that a dialog box template called MyDialog has been

defined. The next step is to define a variable based on that template:

```
dim d as MyDialog
```

The variable d now refers to a dialog box variable based on the template. To make the dialog box appear on the screen, use the DIALOGBOX function:

```
dim result as integer
result = dialogbox(d)
```

The variable result will contain the ID of the button used to close the dialog (this can be used to determine whether the user pressed OK or Cancel, for example).

Normally a dialog box will appear with all fields blank and all check boxes and radio buttons turned off. If you would like to "preload" the dialog box with certain information, just assign the data to fields in the dialog box just as with a user defined type:

```
dim d as MyDialog
d.name = "John Doe"
call dialogbox(d)
```

In the above example the name field will be preloaded with the string "John Doe". After the dialog box has been executed, the name field will contain whatever the user placed in the corresponding edit field on the screen.

Using Dialog Box Response Functions

For more flexibility in creating dialog boxes, *QmodemPro for Windows* allows "dialog box response functions" to be declared. These are functions in the script program that are executed when certain events occur while the user is using the dialog box.

This is the syntax for a dialog box response function declaration:

```
FUNCTION dialogtype.{fieldname | ID(id)} AS INTEGER
   ...
END FUNCTION
```

Dialog box response functions can respond to the following types of events:

The changing of a check box or radio button

The changing of a combo box or list box selection

The changing of an edit text field

The pressing of a button or default button

To define a response function for the first three types of events, use the following form of the declaration:

```
FUNCTION dialogtype.fieldname AS INTEGER
   ...
END FUNCTION
```

This function will be called whenever the associated field changes. Each of the fields in the corresponding dialog variable are available to this function directly without the need to refer to the actual dialog variable. For example, assume the following dialog box declaration:

```
dialog mydialog 50, 50, 150, 50
```

```
      caption "Test dialog"
      username as edittext 101, 20, 10, 110, 12
      guest as checkbox "Use GUEST account", 102, 20, 30, 110, 10
   end dialog
```

Suppose we want to define a response function for the "guest" checkbox that automatically modifies the "username" field. Declare the response function like this:

```
function mydialog.guest as integer
   if guest then
     username = "Guest"
   else
     username = ""
   end if
end function
```

Now, whenever the "guest" button is turned from off to on, the "username" field will be automatically modified to contain the string "Guest". Whenever the "guest" button is turned from on to off, the "username" field will be cleared.

To define a response function for a button, you must use the following syntax for the response function declaration:

```
FUNCTION dialogtype.ID(id) AS INTEGER
   ...
END FUNCTION
```

This function will be called whenever the button associated with the given id is pressed.

Each of these dialog response functions returns an integer. This integer is used by the dialog manager in the following way: If the return value is nonzero, the dialog is closed and the DIALOGBOX function returns the value returned from the dialog box function. If the return value from a response function is zero, the dialog will not be closed.

Unlike a normal function, the return value from a dialog box response function is activated by assigning to the special variable DIALOGRESULT within the function. This is instead of assigning to the name of the function as in a regular user defined function.

The Dialog Init Function

There is one more special dialog response function that is called during dialog initialization before the dialog box is displayed. This dialog response function is declared like this:

```
SUB dialogtype.INIT
   ...
END SUB
```

This function can be used to initialize any fields within the dialog, or to fill a list box or combo box with items. Note that this is the first place during dialog box creation that you are allowed to fill a list box or combo box with items.

To fill a list box or combo box with items, use the AddListBoxItem or AddComboBoxItem subroutines like this:

```
call AddListBoxItem(HWindow, 102, "choice 1")
```

In this example, HWindow is a special variable available within a dialog box response function that refers to the Windows dialog box. 102 is the ID value for the list box that we are working with, and the "choice 1" string is the string to add to the list box. The

AddComboBoxItem works similarly.

Since the HWindow identifier used in this function is really a hidden data member of the dialog structure, the AddListBoxItem and AddComboBoxItem subroutines can only be called from the dialog box INIT function or a dialog box response function.

Using DLL Functions

Microsoft Windows provides special libraries of routines called dynamic-link libraries (DLLs) that let applications share code and resources. *QmodemPro for Windows* script language allows you to call functions that reside in external DLLs.

In order to call a function within a DLL from a script, the following rules must be observed:

The function must use the Pascal calling convention. This means that the name of the function is case insensitive and is not preceded by an underscore character. Also, the parameters to the function are pushed from left to right and are removed from the stack by the function, not the caller.

The function cannot have parameters passed by value other than INTEGER, LONG, STRING, or a pointer to a user defined type.

If the function returns a value, it must be of type INTEGER, LONG, or STRING.

Declaring DLL Functions

DLL functions must be declared using the following syntax of the DECLARE statement:

```
DECLARE {SUB | FUNCTION} name LIB "libname" [ALIAS "aliasname"]
[(arglist)] [AS type]
```

The keyword LIB indicates the name of the DLL in which the function resides. This keyword must be followed by a literal string that names the DLL (the .DLL extension need not be supplied).

Normally, the function is expected to be in the DLL under the same name as the function is declared in the DECLARE statement. In some cases this may be undesirable, as in the case where the name of the function in the DLL duplicates a reserved word or an identifier already declared in your script. In this case the ALIAS clause allows you to use a different name for the function in the script language DECLARE statement, but the alias name is the actual name of the function in the DLL.

Calling DLL Functions

Once a DLL function is declared, it can be called just like any other subroutine or function. For example, to use the MessageBox function, declare it as follows:

```
declare function MessageBox lib "user" (hwnd as integer, message as
string, caption as string, flags as integer) as integer
```

Once declared as above, you can use the MessageBox function like this:

```
dim i as result
result = MessageBox(0, "Save changes?", "Question", MB_YESNOCANCEL)
```

The above example works provided that MB_YESNOCANCEL is defined to be the same as its value in the WINDOWS.H file supplied with the Windows SDK.

In some cases, you may not care about the return value of a DLL function. Instead of using a dummy variable to accept the return value, you can eliminate the return value entirely by declaring the DLL function as a SUB instead of a FUNCTION.

Parameters Passed to DLL Functions

The following specific rules are used when passing parameters to DLL functions:

INTEGER and LONG -- passed by value instead of reference.

STRING -- a pointer to a null terminated string is passed.

user defined type -- a pointer to the user defined type is passed.

When returning an INTEGER or a LONG from a DLL function, the standard parameter passing rules are used. When returning a STRING, the function is expected to return a far pointer to a null-terminated string. *QmodemPro for Windows* will automatically make a copy of the string for the return value.

Compiler Error Messages

When the compiler detects an error compiling a script, it will automatically open up the *QmodemPro for Windows* editor and place the cursor at the position the error occurred. An error message will appear in a dialog box. The error messages that can occur are listed below in numerical order.

Error 1: Error opening input file

Error 2: Error creating output file

Error 3: Syntax error

Error 4: End of statement expected

Error 5: Number too large

Error 6: Error in floating point number

Error 7: String not terminated

Error 8: = expected

Error 9: (expected

Error 10:) expected

Error 11: , expected

Error 12: Type mismatch

Error 13: THEN expected

Error 14: ELSE not allowed here

Error 15: ELSEIF not allowed here

Error 16: UNTIL or WHILE expected

Error 17: LOOP not allowed here

Error 18: EXIT DO not within DO ... LOOP

Error 19: EXIT FOR not within FOR ... NEXT

Error 20: DO, FOR, SUB, or FUNCTION expected

Error 21: Duplicate label declaration

Error 22: LOOP expected

Error 23: END IF expected

Error 24: WEND expected

Error 25: Variable expected

Error 26: WEND not allowed here

Error 27: Function expected

Error 28: Subroutine name expected

Error 29: FUNCTION or SUB expected

Error 30: Identifier expected

Error 31: Type character illegal for SUB

Error 32: AS expected

Error 33: Type name expected

Runtime Error Messages

There are certain errors that cannot be detected at compile time and can occur only while running a script.   These errors will pop up a box on the screen with the error message and terminate running the script. Some of these errors can be caught with the CATCH clause, while some cannot.

"Invalid opcode"

"Unexpected end of file"

"STOP encountered"

"RETURN without GOSUB"

"Math error"

"Array subscript out of bounds"

"Library not found"

"Function not found"

"Invalid file number"

"Error opening file"

"Path not found"

"Error renaming file"

"Communications timeout"

"WHEN ... GOTO executed from noncurrent scope"

"Invalid opcode"

Catch name: (cannot be caught)

This error means that there was a problem with the script's .SCC file and the script probably needs to be recompiled.

"Unexpected end of file"

Catch name: (cannot be caught)

This error means that there was a problem with the script's .SCC file and the script probably needs to be recompiled.

"STOP encountered"

Catch name: (cannot be caught)

This error means that a STOP statement was encountered while running the script.

"RETURN without GOSUB"

Catch name: (cannot be caught)

This error means that a RETURN statement was encountered without a corresponding GOSUB statement.   Be sure that the GOSUB and RETURN statements in your script are properly balanced.

"Math error"

Catch name: <u>ERR_MATH</u>

This error indicates that an arithmetic error has occurred (such as division by zero, taking the logarithm of a nonpositive number, and taking the square root of a negative number).

"Array subscript out of bounds"

Catch name: ERR_ARRAYSUBSCRIPT

An attempt was made to access an array element outside the bounds of the array.

"Library not found"

Catch name: ERR_LIBRARYNOTFOUND

An attempt was made to call a function in an external DLL that could not be loaded.

"Function not found"

Catch name: <u>ERR_FUNCTIONNOTFOUND</u>

An attempt was made to call a function in an external DLL and the named function could not be found.

"Invalid file number"

Catch name: <u>ERR_INVALIDFILENUMBER</u>

An attempt was made to access an invalid or unused file number in a file access statment such as PRINT or GET.

"Error opening file"

Catch name: <u>ERR_FILEOPEN</u>

The file named in an OPEN statement could not be opened (most likely, the file could not be found on disk).

"Path not found"

Catch name: <u>ERR_PATH</u>

This error occurs when a CHDIR, CHDRIVE, MKDIR, or RMDIR statement accesses a drive or directory that does not exist.

"Error renaming file"

Catch name: <u>ERR_FILERENAME</u>

An error occurred during a file rename operation - either the original file does not exist or the destination name is already in use.

"Communications timeout"

Catch name: <u>ERR_TIMEOUT</u>

The time limit has expired on a WAITFOR statement.

"WHEN ... GOTO executed from noncurrent scope"

Catch name: (cannot be caught)

If a WHEN statement contains a GOTO statement, then the destination of the GOTO statement can only be within the current scope (ie. within the same subroutine or function).   Since a WHEN can be triggered at any time, this error can occur if you set up a WHEN statement with a GOTO within it, and then call another subroutine, but the WHEN statement is triggered while in the subroutine.

To avoid this situation, you can set up a global flag that the WHEN statement will set, and then test for this flag during your script. This prevents directly executing a GOTO from within a WHEN statement.

Error 1: Error opening input file

This error occurs when the script compiler cannot open an input file, whether it is the original file or an included file.

Error 2: Error creating output file

This error occurs when the script compiler cannot create the script output file for some reason.

Error 3: Syntax error

This error indicates a generic problem with the script program. Check with the documentation for the statement you are trying to use to make sure the statement is formed correctly.

Error 4: End of statement expected

This error can occur if you have placed extra information on a line that is not part of the statement on that line.

Error 5: Number too large

Integer numbers must be in the range 2147483648 to 2147483647.

Error 6: Error in floating point number

The script compiler has encountered a number which it determines should be a floating point number, but it is not in a correct format.

Error 7: String not terminated

A double quote character appears on the line without a closing double quote character.

Error 8: = expected

An equals sign is expected here. You may get this error if you misspell a script statement -- the script compiler will not recognize the word and may assume that it is a variable you are trying to assign a value to.

Error 9: (expected

An open parenthesis is expected here. You may be trying to call a function with no arguments when it actually requires arguments.

Error 10:) expected

A close parenthesis is expected here. One of the common places where this error occurs is if you omit the CALL keyword when calling a subroutine, but try to place parentheses around the parameter list anyway. This error will occur at the end of the first argument.

Error 11: , expected

A comma is expected here. You may be trying to call a function with fewer parameters than it requires.

Error 12: Type mismatch

A type mismatch error indicates that you are using a variable or expression that is the wrong type for the expression expected there.

Error 13: THEN expected

The THEN keyword is expected after the condition in an IF statement.

Error 14: ELSE not allowed here

The ELSE keyword is only permitted after an IF or ELSEIF clause. If an ELSE clause for the current IF statement has already been encountered, another ELSE clause is not permitted.

Error 15: ELSEIF not allowed here

The ELSEIF keyword is not permitted without a preceding IF statement, and it is also not permitted after an ELSE clause.

Error 16: UNTIL or WHILE expected

The UNTIL or WHILE keyword is expected after a DO or LOOP keyword.

Error 17: LOOP not allowed here

The LOOP keyword is only permitted after a matching DO statement.

Error 18: EXIT DO not within DO ... LOOP

The EXIT DO statement must occur within a DO ... LOOP construct.

Error 19: EXIT FOR not within FOR ... NEXT

The EXIT FOR statement must occur within a FOR ... NEXT loop.

Error 20: DO, FOR, SUB, or FUNCTION expected

The EXIT keyword must be used with one of the listed keywords.

Error 21: Duplicate label declaration

The label you are trying to define has already been defined within the current function.

Error 22: LOOP expected

The LOOP keyword is expected at the end of a DO ... LOOP.

Error 23: END IF expected

The END IF keywords must follow an IF statement.

Error 24: WEND expected

Each WHILE statement must match up with a corresponding WEND statement.

Error 25: Variable expected

A variable is expected wherever an assignment occurs.

Error 26: WEND not allowed here

Each WEND statement must have a corresponding WHILE statement.

Error 27: Function expected

This error means that you tried to call a subroutine as a function. Since subroutines cannot return a value, this is not possible.

Error 28: Subroutine name expected

A subroutine name is expected after the CALL keyword.

Error 29: FUNCTION or SUB expected

Either the FUNCTION or SUB keyword is expected after the DECLARE keyword.

Error 30: Identifier expected

An identifier is expected here. This generally means that you are trying to declare an identifier but none was found.

Error 31: Type character illegal for SUB

Subroutine declarations, since they do not return a value, cannot have a type specification.

Error 32: AS expected

The AS keyword is expected here. Since you need an AS keyword, the preceding identifier needs a type declaration.

Error 33: Type name expected

A type name is expected after the AS keyword.

Error 34: Type specification illegal

A type specification is not allowed in the indicated case.

Error 35: Parameter list type mismatch

A function declaration and the corresponding definition must match in the number and type of arguments.

Error 36: Parameter list has fewer items than declaration

A function declaration and the corresponding definition must match in the number and type of arguments.

Error 37: Parameter list has more items than declaration

A function declaration and the corresponding definition must match in the number and type of arguments.

Error 38: Nested FUNCTIONs or SUBs not allowed

User defined subroutines and functions must not be declared without properly ending the previous subroutine or function.

Error 39: EXIT FUNCTION/SUB not allowed in main code

EXIT FUNCTION and EXIT SUB can only appear within a subroutine or function, not the main program body.

Error 40: AS not allowed here

This error is caused by using the AS keyword in a place where it is not permitted.

Error 41: Duplicate definition

The subroutine or function you are trying to define has already been defined once before.

Error 42: Duplicate identifier

The identifier you are trying to declare has already been used as a variable, constant, subroutine or function name, or type name.

Error 43: Constant expression required

A constant expression is required here. For example, when declaring an array, the array dimension must be a constant.

Error 44: Array variable expected

This error is most often caused by trying to use a subscript with a variable that has not been declared as an array.

Error 45: Integer expression required

An integer expression is required here. For example, when dimensioning an array, an integer value is required for the dimension value.

Error 46: Invalid array bounds

The lower bound of an array must be less than or equal to the upper bound.

Error 47: Reserved for future use

Error 48: File mode expected

One of the keywords INPUT, OUTPUT, APPEND, BINARY, or RANDOM must follow the FOR keyword in an OPEN statement.

Error 49: Reserved for future use

Error 50: Reserved for future use

Error 51: LEN expected

The LEN keyword is expected at the end of an OPEN statement.

Error 52: TO expected

The TO keyword is expected after the starting value in a FOR statement.

Error 53: NEXT expected

The NEXT keyword is expected after a FOR loop.

Error 54: NEXT not allowed here

The NEXT keyword must be preceded by a matching FOR statement.

Error 55: NEXT without FOR

You have used a NEXT keyword with a variable that does not match the variable used in the corresponding FOR keyword.

Error 56: CASE expected

The keyword CASE is required after the keyword SELECT.

Error 57: END SELECT expected

The END SELECT keywords must follow a SELECT CASE block.

Error 58: CASE not allowed after CASE ELSE

The CASE ELSE clause in a SELECT CASE statement must be the last case clause in the statement.

Error 59: ELSE or IS expected

If a CASE keyword in a SELECT CASE statement is not followed by an expression, it must be followed by an ELSE or IS keyword.

Error 60: Statements not allowed here

Statements are not allowed between a SELECT CASE statement and the first CASE clause. Also, statements are not permitted within a user defined type declaration or a dialog box template definition.

Error 61: =, <>, <, <=, >, or >= expected

In a CASE clause within a SELECT CASE statement, the keyword IS must be followed by one of the relational operators listed.

Error 62: END SUB or END FUNCTION expected

The END SUB or END FUNCTION keywords are expected here.

Error 63: EXIT FUNCTION expected

If the EXIT keyword appears within a function, it should be an EXIT FUNCTION statement.

Error 64: EXIT SUB expected

If the EXIT keyword appears within a subroutine, it should be an EXIT SUB statement.

Error 65: END FUNCTION expected

The END FUNCTION keywords must follow a function declaration.

Error 66: END SUB expected

The END SUB keywords must follow a subroutine declaration.

Error 67: Undeclared identifier

Identifiers must be declared before using them. This error is most often caused by using a variable before having a DIM statement that declares it.

Error 68: END TYPE expected

The END TYPE keywords must follow a user defined type declaration.

Error 69: Duplicate type definition

The type name you are trying to declare has already been declared.

Error 70: Type character not allowed

A type specification character is not allowed here.

Error 71: User defined type variable expected

You are trying to use the dot notation to access a field of a user defined type, but the variable to the left of the dot is not a user defined type variable.

Error 72: Field name expected

A field name must appear after a period when accessing a particular field of a user defined type.

Error 73: Dynamic strings not allowed in user defined types

Variable sized strings are not permitted in a user defined type declaration. Use a fixed length string instead.

Error 74: Parameter required

This error occurs when you call a function that requires more parameters than you have supplied.

Error 75: Default arguments not allowed in function definition (use DECLARE)

Default arguments to function can only appear within DECLARE statements, not the actual function body definition.

Error 76: WHEN condition expected

The WHEN condition must be one of MATCH, QUIET, or TIME.

Error 77: DO expected

The keyword DO is required after the condition in a WHEN statement.

Error 78: END WHEN expected

The END WHEN keywords must follow a WHEN statement.

Error 79: END DIALOG expected

The END DIALOG keywords must follow a dialog type declaration.

Error 80: Control type expected

You cannot declare fields of a normal type within a dialog box definition; all fields must be dialog box control types.

Error 81: Dialog variable expected

The DIALOGBOX function requires a variable declared based on a dialog box template as an argument.

Error 82: Field variable required with this control type

An associated field variable is required with the CHECKBOX, COMBOBOX, EDITTEXT, LISTBOX, or RADIOBUTTON control types.

Error 83: Field variable not allowed with this control type

An associated field variable is not permitted with the CTEXT, DEFPUSHBUTTON, GROUPBOX, LTEXT, PUSHBUTTON, or RTEXT control types.

Error 84: Dialog type expected

When declaring a dialog box response function, the name of the type used in the function name must be the name of a dialog box type.

Error 85: Control ID number expected

A control ID number is required when declaring a response function for an unnamed control.

Error 86: Dialog event must be a function returning integer

Dialog response functions must return an integer value to the dialog box manager so that it can tell whether or not to close the dialog box.

Error 87: Dialog init must be a subroutine

The dialog initialization function must be a subroutine -- it cannot return a value to the dialog box manager.

Error 88: String required after ALIAS

A literal string is required after the ALIAS keyword in a DLL function declaration.

Error 89: String required after LIB

A literal string is required after the LIB keyword in a DLL function declaration.

Error 90: LIB only allowed with DECLARE

The LIB keyword is only allowed within a function declaration using DECLARE. You have probably omitted the DECLARE keyword.

Error 91: Function result must be numeric or string

Function results can only be a numeric or string type, not a user defined type.

Error 92: Fixed length string parameter not allowed

Fixed length string parameters to subroutines and functions are not permitted. Use a variable length string instead.

Error 93: Function already defined

The function you are trying to define has already been defined.

Error 94: CATCH not allowed here

A CATCH statement is only permitted at the end of a subroutine or function (or at the end of the main program body). A CATCH statement is not permitted after a CATCH ALL statement.

Error 95: Label "label" not declared in block

You have used the GOTO or GOSUB statement with a label that was not declared within the same statement block as the GOTO or GOSUB statement.

Error 96: Function "function" not declared in script

You have declared a function but have not provided a definition for the function.

Error 97: INPUT or OUTPUT expected

The keyword INPUT or OUTPUT or both is required after the FLUSH command.

Error 98: Positive control ID required for this control type

Control types such as combo boxes and edit text fields require positive control IDs.   See the section on Using Dialog Boxes in chapter 2 for more information.

Error 99: ENTRY, GROUP, MANUAL, or SEARCH expected

One of the above keywords is required after the DIAL command.

Error 100: No local variables can be used in a WHEN block

Because WHEN statements are active until explicitly cleared, you cannot use any local variables of an enclosing procedure because the procedure may exit while your WHEN statement is still active.   You may access global variables (variables declared outside any subroutine or function) from within an WHEN block.

The SLIQ Debugger

Starting the SLIQ debugger is done by selecting the **Script/Debug** menu choice, which displays the standard file selection dialog box. Once a script file is selected, it is recompiled regardless whether an executable **.SCC** file exists. During this compile session an additional debug info file is created in the scripts directory. The debug info file uses the extension **.SCD** and contains references to other files called by the script, line number offsets into the executable file, and type and location information on variables. The debug info file is used only by the debugger and may be erased after debugging is completed.

Once the compile session is complete the debug screen is displayed. It consists of two main sections, the source window and the watch window.

Both the source and watch windows can be arranged and resized within the debugger window, but they cannot be placed outside of the window. If the watch window is not being used it may be minimized, but it cannot be closed.

The Source Window

The source window displays the original source script file. In the left hand margin of the window there is an area where two special symbols are displayed:

An arrow in the left hand column indicates that the current execution position of the script is stopped on the indicated line.

A red circle with a B inside it indicates that the associated line has a breakpoint set on it. Breakpoints are discussed below.

The Watch Window

The watch window is used to watch the values of variables while your script program is executing. If you watch a variable in this window using the Watch button on the toolbar, its value will be updated in the window whenever it is modified by the script program.

The Step Button

The Step button causes the debugger to execute one script statement. If the script statement is a subroutine or function call, the debugger will stop on the first statement of the subroutine or function body.

The Jump Button

The Jump button is similar to the Step button, except that if the current statement calls a subroutine or function, the statements within the subroutine or function will automatically be skipped.

The Return Button

The Return button causes the debugger to execute statements continuously until the return from the current subroutine or function. This button is useful if you Stepped into a subroutine or function and meant to Jump over it.

The Go Button

The Go button continuously executes the script until the script ends or a breakpoint is

encountered.

The Stop Button

The Stop button can be pressed at any time during script execution to cause the debugger to stop the script program at the next possible point.   Note that the script may not stop immediately, but it will stop when the currently executing script statement has finished.

The Watch Button

The Watch button is used to select a variable to watch in the Watch window. If the cursor in the Source window is positioned over a variable name when the Watch button is pressed, that variable will automatically be watched. If not, a selection box will be presented that will allow you to either pick a variable to watch or type in a variable name.

Setting Breakpoints

To set a breakpoint in the Source window, click on the desired line using the right mouse button. To clear the breakpoint, click on the line with the right mouse button again. Note that the breakpoint indicator may not appear on the same line that you click on   it will appear on the next line of the source that contains an executable statement.

Technical Overview of SLIQ

General Program Information

Every variation of BASIC is slightly different, and SLIQ is no exception. This section provides information about the various items that make up SLIQ and how they operate. It only deals with the underlying operational characteristics of the language, not the command set. The entire set of SLIQ commands are discussed in the chapter that follows.

Tokens

Tokens are the smallest meaningful units of text in a *QmodemPro for Windows* script program. There are five kinds of tokens: special symbols, numbers, keywords, identifiers, and string constants.

Special Symbols

Besides letters, numbers, and spaces, SLIQ accepts the following special characters and pairs of characters:

```
 + - * / = < > . , ( ) : ; ' " <= >= <>
```

Numbers

SLIQ accepts numbers in ordinary decimal format. Hexadecimal numbers are represented using 0x or &H as a prefix to the number. Engineering notation (e or E, followed by an exponent) is read as times ten to the power of in real numbers. For example, 7E-2 means 7 x 10-2.

Keywords

Keywords appear in lower case `typewriter font` throughout this section. The following is a complete list of SLIQ keywords:

```
access alias all and append as binary call caption case catch close
const declare dial dialog dialogbox dim do else elseif end eqv exit
flush font for function get gosub goto if imp include input is len let
lib lock loop mid mod name next not open or output print put random
read receive rem return seek select send shared static step sub then
to type unlock until wend when while write xor
```

These keywords cannot be used as identifiers. Since SLIQ is not case sensitive, these keywords may appear in any combination of upper or lower case in your scripts.

Identifiers

Identifiers denote constants, types, variables, subroutines, functions, and fields in user defined types. An identifier can be of any length, but shorter identifiers are usually easier to manage. An identifier must begin with a letter and can not contain spaces. Letters, digits, and underscore characters are allowed after the first character. Like keywords, identifiers are not case sensitive.

String Constants

A character string is a sequence of zero or more characters from the extended ASCII character set, written on one line in the program and enclosed by double quote

characters ("). Case is significant in string constants; "Hello" is a different string from "HELLO".

Comments

Comments consists of text the compiler should ignore, but are useful for annotating your SLIQ programs. Comments are preceded by either the `rem` statement or the   character (single apostrophe).

Note that `rem` is treated as a regular statement, so if you want to place a comment at the end of a line using `rem` , you must precede the `rem` by a colon to separate it from the previous statement. The single apostrophe can be used without the need for a separating colon:

```
print "Hello world!"  'comment OK
print "Hello world!"  :rem comment OK
```

Note that the first needs no colon while the second does.

Constants

A constant is an identifier that represents a value that cant change. A constant is declared in a SLIQ program like this:

```
const Tries = 10
```

This declaration causes the value 10 to be substituted wherever the identifier `Tries` appears in the program. You can also define constants in terms of other constants using expressions, like this:

```
const LookingBad = Tries - 3
```

This will cause the value 7 to be used wherever `LookingBad` appears in the program.

Types

SLIQ is a typed language, which means that each variable or expression has an associated type. This type defines the kind and size of values that the variable can hold, as well as the operations that can be performed on that variable. The following identifiers represent the predefined types in the language:

```
byte
integer
long
real
string
```

Simple types

There are five simple types whose characteristics are shown in the following table:

Type	Range	Format
byte	0 to 255	8 bit unsigned
integer	-32768 to 32767	16 bit signed
long	-2147483648 to 214783647	32 bit signed

real	1.5e-45 to 3.4e38	32 bit IEEE

string   0 to 32767 characters in length   variable length

Strings can also be declared to be of fixed length, which means that they always hold a specific number of characters (even if those characters are just spaces). Here is an example of a declaration of a string of length 30:

```
dim name as string*30
```

In this case `name` will always hold exactly 30 characters. Fixed length strings are particularly useful in user defined types (see below) because variable length strings are not permitted in a user defined type declaration.

Arrays

Arrays are sequences of values, all of which have the same type. For example, the following code declares an array of 10 integers:

```
dim a(10) as integer
```

Actually, there are 11 integers declared here; they are referenced as a(0), a(1), through a(10). The type of elements in an array can be of any type except another array.

User Defined Types

A user defined type is a type that holds more than one value. Unlike an array, the values that a user defined type holds can be of different types. For example, here is a user defined type that can be used to hold a date value:

```
type date
  year as integer
  month as integer
  day as integer
end type
```

In this example all the fields are of the same type. Here is another example that could be used to hold a persons information:

```
type person
  name as string*50
  phone as string*20
  birthdate as date
end type
```

Note that the `date` type declared above can be used inside another type declaration. User defined types can be nested in this way as many levels deep as necessary.

Dialog Box Types

SLIQ supports Windows dialog boxes using a mechanism similar to user defined types. For information on declaring and using dialog boxes, see the separate section on Dialog Boxes at the end of chapter 2.

Variables

A variable is an identifier that represents a value that can change. A variable is declared in a SLIQ program like this:

```
dim count as integer
```

The word count can then be used in any expression where an integer is accepted. To declare an array of values, place the number of values you want to declare in parentheses after the variable identifier. For example,

```
dim friends(10) as string
```

This causes friends(0) through friends(10) to be declared. Declaring a user defined type works the same way:

```
dim bob as person
bob.name = "Bob Smith"
bob.phone = "555-1212"
bob.birthdate.year = 1963
bob.birthdate.month = 4
bob.birthdate.day = 17
```

The above statements also assign values to each of bobs fields. As you can see, we have built upon our previous examples to create a personal record.

Expressions

Expressions are made up of operators and operands. Most operators are binary and take two operands. Two operators are unary and only take one operand. Binary operators use the usual algebraic form (for example, A + B). A unary operator always precedes its operand (for example, -B). There are two unary operators, not and -. Not is a bitwise logical not of its operand, and – takes the negative of its operand. The following is a list of the binary operators in SLIQ, in order of precedence from highest to lowest:

^	exponentiation
* / \ mod	multiplication, division, integer division, modulus
+ -	addition, subtraction
= <> < <= > >=	equality and inequality operators
and	bitwise logical and
or	bitwise logical or
xor eqv	exclusive or, equivalence ((x eqv y) = not (x xor y))
imp	implication, ((x imp y) = ((not a) or b))

Parentheses are used to modify the above precedence order. Expressions within parentheses are always evaluated starting with the innermost set of parentheses. For example,

```
3 + 4 * 5 = 23  (remember, multiplication takes
                 precedence over addition)
(3 + 4) * 5 = 35
```

Functions are called by naming the function to call, perhaps followed by a parameter list

in parentheses after the function name. For example, if s is a string,

```
len(s)
```

is an integer type expression equal to the length of the string s.

Statements

Statements are commands that tell SLIQ what to do. Statements can be broken into two major classes those that control the execution of other statements and those that dont.

Simple statements stand by themselves and dont control the execution of other statements. A simple statement is our first example:

```
print "Hello world!"
```

Compound statements directly control the execution of other statements. Each type of compound statement is described in the following sections:

Single line if statement

The simplest sort of compound statement is the single-line if statement:

```
if x = 5 then print "x is five"
```

This causes the print statement to be executed only if x currently has the value 5. If x does not have the value 5, then the print statement is skipped.

Multiple line if statement

If statements can also span multiple lines, as in this example:

```
if x = 5 then
  print "x is five"
  x = x + 1
  print "x is now six"
end if
```

The style of indenting the lines that are controlled by the if statement is not required, but it makes the program easier to read. You will see this style in each of the following compound statement examples. If statements can also have an else directive:

```
if x >= 1 and x <= 10 then
  print "x is between 1 and 10"
else
  print "x is less than 1 or greater than 10"
end if
```

Finally, if statements can have an elseif directive, which helps make certain constructs easier to write (see the select case statement below for another approach):

```
if x >= 1 and x <= 10 then
  print "x is between 1 and 10"
elseif x > 10 then
  print "x is greater than 10"
else
  print "x is less than 1"
end if
```

Loop statement

The loop compound statement comes in four flavors which differ in when and how the condition to exit the loop is tested. In general, the loop statement will repetitively execute a group of statements until a particular condition is met. For example,

```
x = 1
do while x <= 5
  print "x is "; x
  x = x + 1
loop
```

This is what is called a top-tested loop   the condition is tested at the top of the loop before any of the statements inside the loop are executed. In this case the loop will be executed five times   once for each value of x from 1 to 5. The other kind of top-tested loop is shown below:

```
x = 1
do until x > 5
  print "x is "; x
  x = x + 1
loop
```

The above example functions exactly the same as the first, the difference is the kind of test that is performed to see whether the loop should exit.

Bottom-tested loops work in a similar way, except that the test to see whether the loop should exit occurs after all the statements in the loop. The statements in a bottom-tested loop are always executed at least once (in a top-tested loop the statements may not be executed at all).

```
x = 1
do
  print "x is "; x
  x = x + 1
loop while x <= 5
```

or

```
x = 1
do
  print "x is "; x
  x = x + 1
loop until x > 5
```

An alternate form of the top-tested while loop is shown below:

```
x = 1
while x <= 5
  print "x is "; x
  x = x + 1
wend
```

Select case statement

The select case statement is a convenient way of testing a number of similar conditions. For example, the following statement tests the value of a variable x:

```
select case x
  case 1
    print "x is 1"
  case 2 to 5
```

```
     print "x is between 2 and 5"
   case is > 10
     print "x is greater than 10"
   case else
     print "x is less than 1 or between 6 and 10"
 end case
```

There are several different types of cases that can be tested: single value, range of values, and a relation. An example of each is shown in the above code example. A single case value matches if the case variable is exactly the same as the value given. A range case checks the case variable against the given values to see whether it is within the given range including the endpoints. In the above example, x would be tested as `x >= 2` and `x <= 5`. A relation case is denoted by the keyword `is`, followed by a binary operator, followed by a value. The keyword `is` is replaced by the case variable (x in the above example) and the relation is checked. The `case else` clause is executed if none of the above conditions match.

For statement

The for loop is a convenient way of performing a task a number of times, particularly when the number of iterations is known beforehand. For example:

```
 for i = 1 to 10
   print "i is "; i
 next i
```

In a for loop, the control variable (i in the above case) can be either increasing or decreasing. You may also specify a step value if you want the control variable to change by a value other than one. Here is a loop that prints the odd numbers from 9 to 1 backwards:

```
 for i = 9 to 1 step -2
   print i
 next
```

In the `next` statement that ends a for loop, it is not necessary to name the control variable as in the above example. If the control variable is specified it must match the most recent for loop. For example, the following is illegal:

```
 for i = 1 to 5
   for j = 1 to 10
     print i, j
   next i     ' error, i not last loop
 next j        ' j invalid here as well
```

When statement

The when statement is a structured way of executing statements at the time a particular event occurs. There are a number of different events which can trigger a when statement:

a time duration has elapsed

a certain time of day occurs

a sequence of characters is received from the communications port

The structure of a when statement is very much like the structure of an if statement. For example,

```
when match "press enter" do
  send
end when
```

This will cause the send statement to be executed whenever the characters press enter are received from the communications port. A when statement remains active and can be triggered again until it is explicitly cleared with a `clear when` statement.

Subroutines and Functions

Subroutines and functions can be declared to help perform repetitive tasks. Here is a simple subroutine declaration:

```
sub test
  print "in sub test"
end sub
```

This subroutine can be called in one of two ways:

```
test
call test
```

The call keyword is optional here. Arguments can also be passed to functions, for example:

```
sub test(a as integer, b as integer)
  print "the sum of "; a; " and "; b; " is "; a+b
end sub


call test(1, 2)
call test(6, 11)
test 3, 4 ' notice that parentheses around the
          ' arguments are only allowed with the call
        ' keyword
```

The output of this program is:

```
the sum of 1 and 2 is 3
the sum of 6 and 11 is 17
the sum of 3 and 4 is 7
```

Parameters are normally passed by reference, which means that if a subroutine changes the value of a variable, the change will be reflected in the variable which was used in the original parameter list. The exception to this is if an expression is passed to a procedure. For example,

```
sub bump(a as integer)
  a = a + 1
end sub


dim x as integer
x = 3
print x
bump x
print x
bump x+1
print x
bump (x)
print x
```

The output of this program is 3, 4, 4, 4. In the first call to bump, x is actually incremented because it is directly passed to the variable a in the bump subroutine. In the second call to bump, x is not incremented because it is not directly passed to the subroutine x+1 is passed instead. In the third example, x is also not directly passed because it is enclosed in parentheses.

Functions are similar to procedures, except that they always return a value and this return value can be used in expressions. For example,

```
function f(a as integer, b as integer)
  f = a * b
end function
```

This declares a function which multiplies its arguments and returns their product. It can be used like this:

```
print f(5, 6)
x = y + f(z, w)
```

The first line prints the value 30, and the second line will first multiply z and w, then add that to y, and assign the result to x.