

## **A Guided Tour of QModemPro for Windows 95**

Take a few minutes to get some practical experience with QmodemPro for Windows 95. It's not required, it's just a fast, easy way for you to become familiar with QmodemPro for Window 95 and with Bulletin Board Systems.

Mustang Software, Inc. operates a Wildcat! BBS (Bulletin Board System) to support our customers and provide access to files and messages that can help you use QmodemPro for Windows 95. We call it the MSI HQ BBS.

[Dial the MSI HQ BBS](#)

[Connecting](#)

[Logging on](#)

[A Common Password Problem](#)

[New User Information](#)

[Exploring the BBS](#)

[Downloading](#)

[Expanding Your File](#)

[A Note About Compressed Files](#)

[For More Information](#)

## The Terminal Window

This is the main screen of QmodemPro for Windows 95. The top section of the Terminal window is the Command Menu. Beneath that is the Toolbar. The main screen, below the toolbar, is where all your online data will appear. Beneath the main screen is the Macrobar Menu, and below the Macrobar menu is the Statusbar.

QmodemPro for Windows 95's terminal window is, in a sense, the main control panel used to communicate with your the modem and, once you're on-line, to interact with the remote modem.

This window is surrounded by QmodemPro's main screen which provides information about the status of your connection, a menu bar with command options, a toolbar with buttons for frequently-used functions, and at the bottom of the window, a macrobar and a status line, showing current parameters. You can customize your toolbar or hide it. Click on the toolbar with the right mouse button and choose **Hide** to hide your toolbar or **Properties** to customize it.

## Dialing the MSI HQ BBS

Click on the **Dialing Directory** button near the top left corner of your Qmodem for Windows 95 screen. A new window will open, containing your **Phonebook**.

The QmodemPro for Windows 95 Phonebook is a list of all the Bulletin Board Systems and Online Services you call regularly.

Qmodem for Windows 95 comes with a sample phonebook. The first entry on the list is the MSI HQ BBS. QmodemPro for Windows 95 also includes phonebook entries for some popular online services.

To select the entry you wish to dial, Mustang Software, move the mouse pointer to the entry, and click the left mouse button. When you do that, your selected entry is highlighted — that is, it changes color on your screen.

Now you're ready to dial the entry you just selected. Move your mouse to the **Dial** button near the top left corner of the Phonebook window, and press the left mouse button. You can also double-click on an entry to dial it.

Another window will pop up, showing the name of the system you're calling and details about the entry. These details include how many times you've called, and a status report which shows the number you dialed, and the response from the modem — ringing, busy, connected, and so on.

If the number you're dialing is busy, Qmodem for Windows 95 will pause for a few seconds and try again. The MSI HQ BBS has over two dozen phone lines, so it should only take a few tries to get through if the line is busy on your first try.

Next, we'll [connect to the BBS](#).

## Connecting

When your modem connects successfully with the modem at the other end, you may hear some beeps and squawks as the modems "handshake" or establish a connection. At this point, the modems are exchanging information about the baud rates and connect speeds they support, along with additional information about error correction protocols.

When the modems have completed their handshaking, the phonebook window and status window will disappear, returning you to the terminal window. At this point, the remote system (in this case the MSI HQ BBS) will begin sending data, which you will see on your screen.

Most bulletin board systems have a **command line** interface. That is, they display a menu of available commands, and the user types the letter highlighted on the menu to execute the command.

Next, [log on](#) to the system.

## Logging on

The first screen of information you see on the screen will normally be a brief message identifying the name of the system you've called. You'll then be asked to **log on**.

Logging on means typing your **user ID** (usually your first and last name) at the prompt asking you for this information. Type your first name at the prompt, and press [ENTER] to continue.

You'll then be asked to enter your last name. Type your last name, and once again press [ENTER] to continue.

The BBS will then search its user database for your user ID. Since you're a new user on the BBS, you won't yet be in the database and the BBS will say something like

```
Your name was not found in the user database.  
Is your name spelled correctly?
```

Answer **Yes** to this question if you spelled your name correctly. The BBS will then create a new user record for you, and you'll be asked to select a password.

Think carefully about the password you select. It should be one you can remember easily, but would be difficult for someone else to guess, since this password is your key to the Bulletin Board System. Do NOT use your first name, your spouse's name, or names or words easily associated with you.

The best passwords are a combination of two unrelated words separated by a punctuation mark, for instance

```
RAINBOW$GARBAGE
```

Later on, we'll show you how to enter your password in the phonebook for each system you call, and recall it instantly with a function key.

Sometimes you may be asked for a password and you've never logged on before. [Click here for a possible solution](#), or [continue](#).

### **A common password problem**

Some people with common names (like "Bill Williams" or "Jenny Johnson") may find that someone else with the same first and last names has already set up a user account on the BBS. If you logged on with the same name as a user already on the system, you'll be asked to enter your password. Since you obviously don't know the password, the easiest thing to do right now is hang up and try again. Use the **Hang Up** button on the right-hand side of the toolbar (it looks like a little telephone that's off the hook) to do this.

Since each user name must be unique, you might want to consider using first name plus your middle initial and last name to identify you. To log on with a middle initial, enter your name like this, at the "What is your first name" prompt:

```
Bill M;Williams [ENTER]
```

Note the semicolon after your middle initial — this notifies the Bulletin Board System that the text "Bill M" should be treated as the first name, and the text "Williams" should be treated as the last name.

Next, you will be asked for [New User Information](#).

## **New User Information**

As part of the new user logon procedure, the operators of the BBS may ask you for some demographic information — your mailing address, phone number, and your QmodemPro for Windows 95 registration number. You will find this number stamped on the card bound inside the cover of this manual.

MSI (the makers of QmodemPro for Windows 95 and other communication software) uses this information to determine customer needs and preferences when planning future product releases. The information you provide is confidential, and is not made public.

[Now you can explore the BBS.](#)

## Exploring the BBS

When your new user information has been stored, you will see several welcome screens, with information of interest to MSI customers. After you read these screens, the system will look for any personal mail for you. If there is new mail for you, the system will let you know. You can read your mail now or wait until you've been explored the BBS.

When the BBS finds new mail, it will ask you if you want to (R)ead new mail, (L)ist new mail, or (C)ontinue. Press **R** to read any new mail messages.

Next, you will arrive at the main Bulletin Board System menu. This is the point from which you can select all the options the BBS has to offer: file transfers, e-mail, doors to other applications, and more. Take a minute or two to look at your options in the main menu.

You can access all the other sections from here. Press the letter of the section you want to visit. You may want to change your settings. Type "**Y**" to see and change your settings.

You can enter your personal information, like what type of computer you're using, determine the Help Level that's best for you (for now, Novice is probably the best. Later, you may want to change it to Regular), turn on auto spell checking, and many other options.

The next thing we'll do is check for new mail. If you chose not to read your new mail when you first logged on to the BBS, you can read it at any time. To check for personal mail:

From the **BBS Main Menu**, select **Read All New Personal Mail**. Type **!** (an exclamation point) at the prompt. The system will search all the conferences for any new mail for you.

Downloading a file is easy! Let's [continue](#).



## For More Information

If you're interested in learning more about modem communications and the on-line community, we recommend the following publications:

"Using Computer Bulletin Boards", 3rd edition, by John V. Hedtke (1995, MIS Press, ISBN 1-55828-391-9). This book explains in easy to understand terms how to log on to a BBS, how to read and answer e-mail messages, how to transfer files, and much more. Ideal for the beginner who wants practical, non-technical information.

"Modems for Dummies", by Tina Rathbone (1993, IDG Books Worldwide, Inc., ISBN 1-56884-001-2). This book gives you a good overview of modems and telecommunications basics. It explains in plain, easy to understand language how modems work.

"BBSs for Dummies", by Beth Slick and Steve Gerber (1995, IDG Books Worldwide, Inc., ISBN 156884-900-1). An in-depth, easy to understand book, written in plain English, that explains BBSs, cyberspace, and more.

"Boardwatch" magazine, published monthly by Jack Rickard. Call 1-800-933-6038 for subscription information. A witty, well written, sometimes opinionated overview of BBS issues. Many good BBS lists, by region and theme.

Most cities and towns have a PC User Group. These groups serve the non-professional computer user by providing social and educational meetings, special interest groups, informal classes for beginners, and guest speakers from the computer industry. Most also provide a Bulletin Board System (BBS) for their members.

Your local computer store or public library can provide information on PC User Groups in your area. If you work for a large corporation or government agency, your employer may also sponsor a PC User Group — ask your PC support people for more information.

The on-line community is as close as your nearest Bulletin Board System. To help get you started, we've included a text file containing a list of bulletin board systems using Wildcat! BBS software from MSI. When you installed QmodemPro for Windows 95, it added an icon named "BBS List" in your program menu. Click on the icon to view the file.

Besides the Bulletin Board Systems in your own area, there are several national and international commercial on-line services. Your QmodemPro for Windows 95 package may contain informational brochures on how to subscribe to one or more of these on-line services.

## Downloading

The final part of our on-line tutorial will be to **download** (receive) a file. Downloading means copying a file from the remote computer (the host, or BBS, or Online Service) to your computer over the modem, using a file transfer protocol.

Press **F** at the **Main Menu**. The BBS will then show you the **File Menu**. This menu has a list of selections having to do with file transfers. For instance, you can search for a file, list the names and descriptions of files on the system, or view the contents of a file you're interested in.

We have created a special file for this tutorial. It contains a selection of sample scripts and other useful tools for Qmodem for Windows 95 users. The name of the file is Q95INFO.EXE.

The most important thing to remember about transferring a file is that you need to tell the computers at both ends of the connection what you want to do. The **host** computer (the BBS or Online Service) needs to be told first. You do this by selecting **Download** from the file menu — as with our previous examples, type the first letter of the command, in this case **D**.

The BBS will ask you the name of the file you want to download. Type

```
Q95INFO.EXE [ENTER]
```

The BBS gives you the option to select several files, and download them in a batch. Since we are only going to download one file during this session, press your [ENTER] key again when prompted for the second file. The BBS will then ask you if you would like to logoff automatically when the download has finished. Since we're just about finished with this tutorial, answer **Yes** to this question.

Zmodem, the most commonly used protocol, is used automatically for new users of the MSI HQ BBS. It is also the default protocol used for QmodemPro for Windows 95 phonebook entries. The BBS will begin sending the file, and QmodemPro for Windows 95 will begin receiving it automatically. Below is a picture of the File Transfer Status window, showing information about the file transfer in progress.

We will explain more about each part of this window later on in this book. When the file transfer is complete, the download status window will disappear, and you will once again see the terminal screen.

Since you selected **Logoff after transfer** before you began downloading your file, the BBS will now display a 10-second countdown. This countdown gives you the opportunity to change your mind and stay on-line by pressing a key. For now, let the countdown finish. The BBS will automatically log you off and hang up the phone.

[Expand your new file.](#)

## Expanding your file

The file you just downloaded is a self-extracting archive — that is, it acts as a container for smaller files inside, which unpack automatically when you execute the file. To remove the files from their container, click on the **Explore Download** button on the macrobar. This macro launches Explorer in your download directory.

Drag the file from the **Download** directory to the **Scripts** directory. It will be in the QmodemPro for Windows 95 download directory (C:\PROGRAM FILES\MUSTANG SOFTWARE\QMODEMPRO\DOWNLOAD, if you have used the default settings).

Before closing Explorer, double-click on the QMWIN.EXE file in the SCRIPTS directory to extract the files from their container. Then close Explorer.

Next, from the QmodemPro for Windows 95 main menu, select **Scripts/Execute**. A dialog box will appear, asking you to choose a file. The new script files should appear in the list of files. Double click on a filename, or mark the filename and then click Open.

Use Explorer to delete Q95INFO.EXE. The script file you downloaded will run without this file, once it has been extracted from the container.

Now that we've completed the tutorial, it's time to shut down QmodemPro for Windows 95. Use your mouse to go to the **File** menu, and select **Exit**, or click the (close) button in the top right- corner of the screen.

Click [here](#) for more information about modems, BBSs, and the electronic community.

## Macro System Functions

Macros can be used to perform many functions automatically. These can be assigned to buttons located at the bottom of the terminal window. Click on the macro you want to know more about:

[@ADDLFTOCR \[on | off\]](#)

[@AUTOANSWER \[on | off\]](#)

[@BREAK](#)

[@CAPTURE \[filename | off\]](#)

[@CLS](#)

[@DIAL \[ENTRY | GROUP | SEARCH | MANUAL\] string](#)

[@DIALMANUAL\[NNN-NNNN,AAA,CCC\]](#)

[@DIRECTCONNECT](#)

[@DOORWAY \[on | off\]](#)

[@DOWNLOAD \[filename/dir \[protocol\]\]](#)

[@DUPLEX \[on | off\]](#)

[@EDIT \[filename\]](#)

[@EMULATION \[emulation\]](#)

[@EXIT](#)

[@EXPLOREDOWNLOAD](#)

[@EXPLOREUPLOAD](#)

[@HANGUP](#)

[@HELP](#)

[@HOST](#)

[@MANUALDIAL\[NNN-NNNN,AAA,CCC\]](#)

[@MODEMPROPERTIES](#)

[@PASSWORD](#)

[@PHONEBOOK \[phonebookname\]](#)

[@PRINTER \[on | off\]](#)

[@QUICKLEARN \[scriptname\]](#)

[@RESETEMULATION](#)

[@SCRIPT \[scriptname\]](#)

[@SCRIPTCOMPILE \[scriptname\]](#)

[@SCRIPTDEBUG \[scriptname\]](#)

[@SCRIPTEDIT \[filename\]](#)

[@SCROLLBACK \[on | off\]](#)

[@SCROLLBACKRECORD \[on | off\]](#)

[@SHELL \[filename\]](#)

[@SPLITSCREEN \[on | off\]](#)

[@STAMP \[string\]](#)

[@STRIPHIBIT \[on | off\]](#)

@TRAPSCREEN [filename]

@UPLOAD [filenames [protocol]]

@USERID

@VIEW [filename]

@VIEWPICTURE [filenames]

@XONXOFF [on | off]

**@ADDLFTOCR [on | off]**

This is the same as menu command Connection/Add LF to CR.

**@AUTOANSWER [on|off]**

This is the same command as Auto Answer on/off.

**@BREAK**

This is the same as menu command Connection/Break.



**@CAPTURE [filename | off]**

Toggles the capture file on and off.

**@CLS**

Clears the terminal screen.

**@DIAL [entry|group|search|manual] string**

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

**@DIALMANUAL [NNN-NNNN,AAA,CCC]**

This is the same as the menu command Dial/Manual Dial. N is the phone number, A is the area code, C is the country code. IF you do not specify an area or country code, the dialer will use your default.

## **@DIRECTCONNECT**

Opens a direct connection, bypassing TAPI. Any valid COM port may be specified. If no parameter is specified, a direct connection dialog box is opened. This is the same as the menu command Connection/Direct Connect.

**@XONXOFF [on|off]**

Changes XON/XOFF flow control to specified parameter. Same as menu command Phonebook/Edit Phonebook Entry/Connect Using/ Advanced/Xon/Xoff.

**@VIEWPICTURE [filenames]**

Opens the Picture Viewer and loads in specified filenames.

**@VIEW [filename]**

Opens the text editor and loads in the specified file.



**@USERID**

Extracts the user ID from the USERID field in the current phonebook entry. This is valid only when dialed from the phonebook.

**@UPLOAD [filenames [protocol]]**

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

**@TRAPSCREEN [filename]**

Saves the current screen to the specified filename. This is the same command as menu command Edit/Screen to File.

**@DOORWAY [on|off]**

Same as menu command Connection/Doorway. This command can only be used if the remote system supports it.

**@DOWNLOAD [filename/dir [protocol]]**

Downloads the specified file, using the specified protocol.

**@DUPLEX [on | off]**

Turns full duplex on or off. Same as menu command Connection/Full Duplex.

**@EDIT [filename]**

Opens the Editor and loads the requested file.

**@EMULATION [emulation]**

Sets the emulation as specified. Same as menu command Connection/Emulation.



**@EXIT**

Exits QmodemPro for Windows 95 completely. This command cannot be used to close the phonebook, editor, or picture viewer.

**@EXPLOREDOWNLOAD**

Opens the Windows Explorer in your download directory.

**@EXPLOREUPLOAD**

Opens the Windows Explorer in your upload directory

**@HANGUP**

Sends a hangup command. Same as menu command Connection/Hangup.

**@HELP**

Opens the QmodemPro for Windows 95 Help System in the terminal window.

**@STRIPHIBIT [on |off]**

Turns the 8th bit strip feature on or off. Same as menu command Connection/8th bit strip.

**@STAMP [string]**

Records an entry into the log file.

**@SPLITSCREEN [on | off]**

Turns the split screen terminal view on or off. Same as menu command View/Split Screen.



**@HOST**

Executes the Host Mode script

**@MANUALDIAL [NNN-NNN,AAA,CCC]**

Dials a number manually. N is number, A is area code, C is country code. If no parameters are set, the Manual Dialog Box is opened.

## **@SHELL**

Go to a Command Line prompt and execute the requested application.

**@SCROLLBACKRECORD [on | off]**

Turns the scrollback record option on or off. Same as the menu command View/Scrollback Record.

## **@MODEMPROPERTIES**

Opens a property sheet allowing the configuration of modem properties. This is the same as the menu command Phonebook/Edit Phonebook Entry/Connect Using/Configure.

**@SCROLLBACK [on | off]**

Opens the scrollbar log for viewing or closes it.

**@PASSWORD**

Extracts the Password field from the current phonebook entry. This only works when the number is dialed from the phonebook.

**@SCRIPTEDIT [filename]**

Opens the scripts editor and loads the requested script.



**@PHONEBOOK [phonebookname]**

Opens the requested phonebook. Same as menu command Phonebook/File/Open.

**@SCRIPTDEBUG [scriptname]**

Opens the script debugger and loads in the requested file.

**@PRINTER [on | off]**

Toggles the print capture option on or off. Same as menu command File/Printer Toggle.

**@QUICKLEARN [scriptname]**

Executes the QuickLearn script and saves it with the specified filename.

**@RESETEMULATION**

Resets the current emulation to its default configuration.

**@SCRIPT [scriptname]**

Executes the requested script. Same as menu command Scripts/Execute.

**@SCRIPTCOMPILE [scriptname]**

Compiles the requested script. Same as menu command Scripts/Compile.

## Troubleshooting Guide

Sometimes QmodemPro for Windows 95 may not work for you the way you believe it should. If your troubleshooting efforts are to have any meaning, do not change more than one software setting at a time because you won't be able to tell which new setting actually solved the problem. If you believe the modem is not working properly, check all connections first. Be sure the telephone line is plugged into the wall outlet and the modem. Always turn **off** the power before opening your PC! If the modem was turned off when you turned your computer on, you may need to reboot with the modem turned on in order for Windows 95 to detect it.

If your modem still doesn't work the way it should after you've verified that it is connected properly, refer to your Windows 95 User Guide or the modem manufacturer's documentation for help with modem problems.

Here are some other possible causes:

[Line Noise](#)

[Flow Control](#)

[Serial Ports](#)

[Software Settings](#)

[Transfer Problems](#)

[GP Faults](#)

[Lockups](#)

[Call Waiting](#)



**Line Noise**

Are your telephone lines exceptionally noisy? Noise over the phone line 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.

Did someone pick up an extension phone in your home, while you were on-line? This frequently causes data calls to disconnect abruptly.

## Flow Control

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). Windows 95 defaults to Hardware Flow control. If you wish to change this setting, it can be changed either in the QmodemPro for Windows 95 phonebook properties settings (through **Edit/Entry/Properties/ Advanced**) or in the Windows 95 Control Panel.

## Serial Ports

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.

One possible way to correct this might be to check the FIFO settings in **Modems/Port Settings/Advanced** in the Windows 95 Control Panel. This may data to flow into the processor more rapidly. Refer to the Windows 95 manual for more information about this setting.

## Software settings

Software settings are another possible cause for garbled screens. There are a couple of settings in the QmodemPro for Windows 95 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). These properties must be changed in Windows 95 **Settings/Control Panel/Modems**.

Another possibility is that you've got **8th bit strip** turned **on**. When you're connected to a system that is doing this, select **Connection/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 95 is thus not interpreting the ANSI escape sequences (a lot of [ brackets and numbers) as graphics and color. Try changing the emulation by selecting **Connection/Emulations** and select ANSI.

You should suspect an error in the way you have set up QmodemPro for Windows 95 if you have the same problems on every remote host you call.

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.

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, 95 and if the modem itself if it has flow control settings, check them. Refer to your modem manufacturer's documentation.

Another thing that you might check is the modem properties in the Windows 95 **Settings/Control Panel/Modems**

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.

## **File Transfer Problems**

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.

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 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 Windows 95 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.

If you have a problem, check the system properties in the Windows 95 Control Panel. There could be a conflict causing one of the ports to be disabled. Refer to your Windows 95 documentation, and the manual for your modem, serial card and communication.

If you make changes to the setup, you should change only one thing at a time. This allows you to determine exactly what needs to be changed, and if you make a change that doesn't work, it can be easily corrected.

**GP Fault**

A General Protection or GP fault is only caused by a break or fault in Windows 95, by a driver failure, or a failure in a currently running application, like QmodemPro for Windows 95.

## **Lockup**

If your computer seems to have locked up, press CNTRL+ALT+ DEL. A menu pops up showing all programs currently running. If a program, such as QmodemPro for Windows 95, has locked up or become busy, there will be a message next to that program telling you that the program is not responding. Highlight that program and click the **End Task** button. You will lose any unsaved work in the program, but ending the task will free the rest of the system.

If you are able to make QmodemPro for Windows 95 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.

## Call Waiting

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. You can disable this option by editing either **Edit/Phonebook Entry/Dialing Properties** in the QmodemPro for Windows 95 phonebook or the Windows 95 Control Panel settings. Check the box to toggle the disable option **on**, and choose the appropriate number to dial to disable it.



## **The Phonebook**

The dialing directory, or **Phonebook**, contains 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 mark and organize entries into groups, for multiple dialing sessions.

Each Phonebook stores 1024 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.

[Organizing the Phonebook](#)

[Recording Your Data](#)

[The Connection Property Sheet](#)

[The Options Property Sheet](#)

[The Statistics Property Sheet](#)

[Converting Phonebooks from Another Format](#)

[About modems in QmodemPro for Windows 95](#)

## Organizing The Phonebook

Entries may be shown as large or small icons, or in rows with details about each entry displayed across the screen. If you choose to view by large icons, no details will be available in the window. You must switch views to [show details](#). To change the Phonebook view, click the right mouse button on any unused area of the phonebook and select **View**, or choose the **View** menu. Select the view you want to see.

Like the Terminal window, the Phonebook window has a menu bar and a toolbar. You can change the look of QmodemPro for Windows 95 by clicking the right mouse button on the desktop area (between the terminal window and the frame), and selecting **Properties**.

Click the **New** button on the phonebook toolbar to add new directory entries, or click the right mouse button on an unused area of the phonebook and choose **New**. You can also use the **Edit/New** command from the phonebook menu, or press **INS**. To edit an existing entry, click the right mouse button on the entry and select **Properties**, or choose the **Properties** toolbar button, or from the menu select **Edit/Properties**. The **Edit Phonebook Entry** property sheet will pop up, allowing you to edit or add information about the selected Phonebook entry.

QmodemPro for Windows 95 allows you to set the dialing properties for each entry in your Phonebook. To customize dialing on your specific phone system, just start from the phonebook and click the right mouse button on the entry you want to change and select **Properties**. Enter the information for **My Locations**. Click on **OK** when you are satisfied that the information is correct. Following is an example of the Dialing Properties property sheet:

You can use this information to send long distance access code numbers, PBX codes, extension numbers, and more. Note that these 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 first).

If you need to enter a phone credit card number after dialing a number, you can store the card 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 TAPI will not translate the letters into numbers.

Special control characters may also be entered, if necessary.

## **Connection Property Sheet**

The Connections property sheet contains information used to dial phonebook entries.

[System Name](#)

[Primary Number](#)

[Country Code](#)

[Dialing Properties](#)

[Connect Using](#)

## **Options Property Sheet**

The Options property sheet contains values used during connection:

[Emulation](#)

[File Transfer Protocol](#)

[User ID](#)

[Password](#)

[Notes](#)

[Macros](#)

[Script](#)

[RIPIcon](#)

[Dial Retries](#)

[Translate Tables](#)

## The Statistics Property Sheet

QmodemPro for Windows 95 keeps track of statistics for each entry in the [Statistics](#) property sheet: 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.

## **Recording Your Data**

You may wish to keep a temporary or permanent record of your on-line connection. QmodemPro for Windows 95 provides several ways to capture this data: a [Capture File](#), [Scrollback](#), [Screen to](#) , [Selection to](#) , or a [Log file](#).

## The 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 information as it scrolls across the terminal window, including 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 toolbar 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. The default for this feature is the name you assign to the capture file.

## The Scrollback File

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 **View/Scrollback** or click the **Scrollback** button on the toolbar. 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. The limit can be set by opening the **Tools/Options/Emulations/Properties/General** property sheet and entering a number for Scrollback.



## Screen/Scrollback/Selection to Printer

There are several kinds of data that can be sent to the printer.

Select **Edit/Screen to** and then click on **Printer** to send the contents of the current screen to the printer.

To send the contents of the scrollback area to the printer, choose **Edit/Scrollback to** and then **Printer**.

You can send selected text on the terminal screen to the printer by choosing **Edit/Selection To** and then **Printer**. This can also be done by highlighting your selection, then dragging it to a printer desktop shortcut.

When you choose this method, Print works in the same way as the Capture function, in that the entire on-line session is recorded, in this case to the printer. Click on the **Printer** button on the toolbar or select **File/Printer Toggle** to enable printer toggle and select it again to disable it.

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.

You can also mark the screen or the section of it that you need and drag the information directly to the desktop, to an open file, or to another open application.

## **The Log File**

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

## **Uploading and Downloading**

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

[Batch Uploads](#)

[Transfer Protocol](#)

[Using the File Clipboard](#)

[Canceling A Transfer](#)

## **The Editor**

QmodemPro for Windows 95 has a built-in text editor, which can also be used as a standalone program. For more information on the editor, open the editor and activate the help engine.

Click this button {button QMEDITOR,JI(`QMEDITOR.HLP',`IDH\_CONTENTS')} to find out more about the QmodemPro for Windows 95 Editor.

## The Picture Viewer

You can use the Picture Viewer to look at graphic files. Select the [Tools](#) menu option, and then **Picture Viewer** to pop up the viewer window. From there, you can select one or more GIF, BMP, or JPEG files to view, either one at a time, or multiple files.

If you have turned on the **View Picture** option in your **Tools/Options/Protocols** setup (from the QmodemPro for Windows 95 main menu), QmodemPro for Windows 95 will automatically open the Picture Viewer when you begin downloading a .GIF, .BMP or .JPG file. If the View Pictures option is turned on, a check will appear beside it. You can view the file as it is downloading. The Picture Viewer allows multiple pictures to be viewed consecutively as they are being downloaded. Use the **Tools/Options** menu choice and select the **Protocols** Sheet to check or change your settings.

Click this button {button QMVIEWER,JI(`QMVIEWER.HLP',`IDH\_CONTENTS')} to find out more about the QmodemPro for Windows 95 Picture Viewer.

## 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 uncompress 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) of this and other compression utilities, from MSI HQ BBS and most local Bulletin Board Systems. To use this file, copy it to a directory in your Windows 95 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, such as WinZip or Norton Navigator, work in the same way. If you need help understanding the use of commands such as COPY and PATH, refer to your Windows 95 manual.

## **The Scripts Menu**

There are several options available in the Scripts menu. Click on one of the topics below or refer to the Scripts section of the online help or your SLIQ Script Guide in your documentation for detailed information.

[QuickLearn](#)

[Execute](#)

[Edit](#)

[Debug](#)

[Compile](#)

[Host Mode](#)

[SERVICES.QSC](#)

## **Command Line Access**

To access a command line prompt, click on the DOS selection of the default macrobar, or select **Start/DOS Prompt** from the Windows 95 taskbar.



## Modems

Because [TAPI](#) allows the sharing of COM ports, modems are configured in Windows 95, and placed in a registry. Rather than having to specify all the necessary communication parameters manually, these parameters are associated with a modem. Modems are then associated with Phonebook entries. When you add or edit a phonebook entry in QmodemPro for Windows 95, all you need to do is choose the modem you want to use for that specific phonebook entry.

To choose a modem, go to [File/Dialing Directory/Edit/Properties](#). Click on the down arrow next to the [Connect Using](#) field. A list of available modems will drop down. Click on the modem you want to use, and QmodemPro for Windows 95 takes care of the rest.

Most of the time, the default settings registered in Windows 95 will be correct. If you need to change these settings, simply change the necessary parameters associated with the modem from the **Dialing Directory** menu. Highlight the phonebook entry, select **Edit/Properties**, and edit your **Connect Using** settings. Save your changes, and the work is done!

If you determine that you need to change the default settings for your modem, refer to Chapter 4, Advanced Operations, in your QmodemPro for Windows 95 User Guide for more detailed information.

**QuickLearn**

Records commands sent to and from the keyboard while on line. This is then compiled and automatically creates a script from the recorded session.

## **File Menu Options**

The File menu in the main terminal screen has several options. These options allow you to open a phonebook, begin downloads and uploads, and toggle recording options on and off. The options opened from the File menu are:

[Dialing Directory](#) This command brings up the Phonebook, or dialing directory.

[Upload](#) Opens an upload dialog box.

[Download](#) Opens a download dialog box.

[Capture](#) Toggles the Capture log on and off.

[Log Toggle](#) Toggles the Session log on and off.

[Log View](#) Opens the Session Log for viewing.

[Printer Toggle](#) Toggles the Printer log on and off.

## **Edit Menu Options**

The Edit Menu allows you to manipulate data on and off screen, saving it to other applications, finding text, and viewing screens from the scrollback feature. The options opened from the Edit menu are:

[Copy](#) Copies marked text to the clipboard

[Paste](#) Pastes copied text to the cursor location.

[Screen to](#) Sends data on the current screen to the clipboard, capture file, a named file, the printer, or a mail message.

[Scrollback to](#) Sends data in the scrollback log to the clipboard, capture file, a named file, the printer, or a mail message.

[Selection to](#) Sends marked data to the clipboard, capture file, a named file, the printer, or a mail message.

[Find](#) Searches for specified text.

## View Menu Options

The View Menu controls how your terminal screen looks, and what you can see from it. The following options are available through the View Menu:

**Toolbar** Toggles the toolbar view on and off.

**Macrobar** Toggles the macrobar view on and off.

[Statusbar](#) Toggles the statusbar view on and off.

**Split Screen** Toggles the split-screen view on and off.

[File Clipboard](#) Opens the file clipboard.

**Scrollbar** Toggles the scrollbar view on and off.

**Scrollbar Record** Toggles the scrollbar record feature on and off.

**Clear Scrollback** Clears the scrollbar buffer.

## Connection Menu Options

The Connection Menu gives you the ability to make minor settings corrections and set emulation options. The following options are available through the Connection Menu:

[Emulation](#) Brings up an Emulation Selection dialog box.

**Clear Screen** Clears the current terminal screen.

**Reset Emulation** Resets the current emulation to its default settings.

**Hangup** Sends a hangup command to the remote system, canceling the connection.

**Break** Sends a break command to the remote system.

**Clear Xoff** Clears the XOFF flow control.

**Full Duplex** If this is checked, full duplex is turned **on**.

**Add LF to CR** If this option is checked, a line feed character is added to carriage returns sent.

**8 Bit Strip** Toggles the 8th bit strip option on or off. A check mark beside this option indicates that the 8th bit strip feature is turned **on**.

**Doorway Mode** Turns the doorway mode feature on or off. If this option is checked, the doorway is turned on.

**Auto Answer** Toggles the Auto Answer option on and off. If this option is checked, AutoAnswer is turned **on**.

[Direct Connection](#) Brings up a Direct Port Connection dialog box. This option allows a direct connection to your modem.

**Properties** Displays the properties used by the current connection.

## Scripts Menu Options

The following options are available through the Scripts Menu command.

[QuickLearn](#) This option toggles the QuickLearn option off and on. This allows Qmodem to “learn” a script by recording an online session.

[Execute](#) Brings up a dialog box allowing you to choose and run a script.

[Edit](#) Brings up a dialog box allowing you to choose and edit a script.

[Compile](#) Brings up a dialog box allowing you to choose and compile a script.

[Debug](#) Brings up a dialog box allowing you to choose and debug a script.

[Host Mode Script](#) Turns the Host Mode on, allowing you to accept remote calls. This is actually a mini-BBS.

## **Tools Menu Options**

Several tools can be accessed and customized through the Tools Menu. The options available from the tools menu are listed below.

[Picture Viewer](#) Opens the graphics file viewer.

[Editor](#) Opens the text editor.

[Customize](#) Allows you to customize the toolbar and macrobar.

[Translate Tables](#) Opens a dialog box allowing you to select various translation tables.

**Options** Allows you to set options that customizes the look of QmodemPro for Windows 95



## **Detail View**

In the **Detail** view, the first field, Service, is the name of the service — for example, Mustang Software BBS, GEnie, etc.

The next group of fields contain information about the service; for instance, the phone number and modem to use 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, or you can resize the fields by holding the mouse at the edge of each field.

When a sizer appears, click the mouse button on the sizer, and drag the field to the size you want.

You can also rearrange the entries in large or small views by dragging them to the location you desire.

You can also rearrange the entries in large or small views by dragging them to the location you desire.

## **The Script Editor**

If you choose the [Scripts/Edit](#) command from the main menu to open the Qmodem Editor, the Editor will open with special features used especially for creating and editing scripts.

The primary differences between the two versions of the Editor deal with scripting manipulation. The Script Editor has a built-in Compiler feature, to allow quick compiling while you are editing.

It also offers a Syntax Highlight option. This option allows you to show specific types of script language (syntax) in user-defined colors. This allows you to visually differentiate syntax at a glance. For a detailed list of these features, refer to Book 2 of the QmodemPro for Windows 95 documentation, the SLIQ guide.

**System Name**

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.

You can change the **Icon** associated with the new entry. To see a list of available icons, click **Change Icon**. Some icons are included with QmodemPro for Windows 95. You can also look in other directories for other icons that may be available. Choose the one you want, and press **OK** to accept. This icon will also appear with your entry if you view it from the Windows 95 Explorer or My Computer.

## **Primary Number**

There are five possible entries for telephone numbers. If the system you call has only one telephone number, enter it in the **Primary Number** field. TAPI requires that you add the area code for any country that uses area codes, even for systems in your dialing area. Windows 95 will correctly dial local numbers without using the area code, but the field must be entered in the phonebook.

In rare cases where the alternate number for a phonebook entry may have a different area code, a separate entry must be created for that number. If the alternate dialing numbers all have the same area code, you do not need to make any additional entries.

## **Country Code**

To see a list of available country codes, highlight the **Country Code** section and press the first letter of the country you want. A list of available countries (beginning with that letter) will appear. Or you can click on the down arrow to the right of the dialog box. Select the appropriate country for the entry.

## **Dialing Properties**

**Dialing Properties** is used to modify Windows 95 settings used to configure alternate dialing locations and any special dialing instructions for each location. It includes support of international dialing and credit card billing numbers.

By selecting **Phonebook/Properties/Edit Dialing Properties**, you can add billing information, special dialing codes, and other information about your phone system.

## **Connect Using**

The **Connect Using** field allows you to select which modem to use for the phonebook entry. QmodemPro for Windows 95 uses modems that are registered in Windows. Click on the **Connect Using** down arrow for a list of registered modems, and select your choice. **Configure** lets you access and modify Windows 95 settings for the modem you've selected.

One field in the **Connect Using** section is **Type**. Choose the type of communication you want to use for the phonebook entry. If you will be calling a BBS, an online service, or another type of data-based connection, choose **Data**. If you want to use the QmodemPro for Windows 95 phonebook to dial a voice line, select **Voice** type. One possible advantage of using QmodemPro for Windows 95 to dial for you is the automatic display dialog box, which shows you the length of your call, allows you to hang up, and can log all the information to a log file. The **Telnet** connection type should be selected exclusively for telnet connections.

## **Emulation**

Your PC emulates, or imitates, a terminal during a communications session, and many different types of terminals are supported by Windows 95. **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.



## **File Transfer Protocol**

Select the default **File Transfer Protocol** you wish to use with this system. In almost all cases, the best choice is **Zmodem**. You'll find a more detailed discussion of transfer protocols later on in the QmodemPro for Windows 95 User Guide.

**User ID**

**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 @USERID macro command. This command is useful when creating logon scripts. See Chapter 4, Advanced Operation for more information on macros.

## **Password**

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 @PASSWORD 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.

## Notes

QmodemPro for Windows 95 allows you to attach a file to a phonebook entry. This is usually a text 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. You can create your notes using the QmodemPro for Windows 95 Editor, Windows 95 Notepad, or any plain text editor.

QmodemPro for Windows 95 uses the Notes file in a special way. When you attach a note file to a phonebook entry and use the **Find** 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**

Macros is the name of the Macro file associated with this entry. The Macro file contains a set of macro assignments for the buttons below the terminal window. This option allows you to customize your button bar for each system.

**RIP Icons**

**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.

## **Dial Retries**

**Dial Retries** tells QmodemPro for Windows 95 how many times to redial an entry if it does not connect on the first call. Set this value to 0 (zero) for continuous dialing until a connection is established, or until you manually abort the redial. To change the number of **Dial 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.

## Scripts

**Script** is the name of the script file QmodemPro for Windows 95 will execute whenever you dial and connect with this Phonebook entry. If the script file does not exist on disk, QmodemPro for Windows 95 will create one for you automatically if a script name is entered in this field, in [QuickLearn](#) mode. QuickLearn is a feature that allows QmodemPro for Windows 95 to learn a simple script by remembering the keystrokes used in a specific sequence.



## **Translate Tables**

Some terminal emulations use special keymaps and characters. These are generally stored as translation tables to allow an emulation to interact with the host system correctly. [Translate Table](#) allows you to choose translation tables. These are not needed by most users.

## QuickLearn

The **QuickLearn** Script Recorder allows QmodemPro for Windows 95 to learn repetitive tasks, such as how to log onto a BBS or Online 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.

There are four 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 click on **Properties** for 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 95 will automatically record a script file.

The third way to launch QuickLearn is to click on the **Learn** macro available on the default macrobar.

You can also use a toolbar icon to launch QuickLearn, if you have configured the toolbar to include this icon. To add a QuickLearn icon, click **Tools/Customize** and select Toolbar, or click the right mouse button on the toolbar and select Properties. Highlight the QuickLearn option in the Available Buttons section, and double click or click Add to add it to Toolbar Buttons. The icon will be added to the toolbar.

After "learning", many QuickLearn scripts are ready to run without further editing or modification. If your newly-learned script requires editing or debugging, use the [Edit](#), [Compile](#), or [Debug](#) feature of the Scripts menu.

## **Execute**

To start running a script file, use the Execute Script button or the **Scripts/Execute** command. This command allows you to launch a script at any time.

**Debug**

Once a script has been compiled, loading a script into the debugger allows you remove any runtime errors from your script. You can also use the debugger to correct incorrect syntax.

## **Compile**

Some scripts must be compiled before they can be run. The **Compile** command automatically compiles the script file you request. If there is an error in the script file, the compiler will load the script file into the editing window at the point of error.

## **Host Mode**

Selecting the [Host Mode Script](#) starts the script file that sets up your host mode. The host mode allows other computers to dial in to your machine.

## **Edit Script**

To edit or create existing script files, choose **Edit** and select the script you wish to change. The script will be opened in the Script Editor.

## SERVICES.QSC

The QmodemPro for Windows 95 default phonebook provides you with icons for several of the most popular online services, ready to go.

If your favorite online service is not already in the phonebook, or if your phonebook entry was mistakenly removed, you can set up an automatic script file with SERVICES.QSC. This is a special script provided with QmodemPro for Windows 95 that automatically creates phonebook entries with logon scripts for a number of online services. To use this script, select **Scripts** from the QmodemPro for Windows 95 menu bar.

Select Execute, and in the directory dialog box that pops up, select SERVICES.QSC. QmodemPro for Windows 95 will compile the script automatically, and put a dialog box on the screen:

Select the service you would like to add, and fill in the fields in the next dialog box. We will use CompuServe as an example — the others are similar in layout.

Enter your CompuServe User ID or “PIN”, your password, and your local CompuServe phone number, if you know it. You can look it up by selecting the **View Numbers** button and searching for your city.

When you have finished filling in the information on the screen, select **OK**. QmodemPro for Windows 95 will create a phonebook entry for you automatically, with an auto-executing script that will log you into your online service.



## 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 (if you haven't selected one already), then type the names of the files you want to download.

You can also use the QmodemPro for Windows 95 [File Clipboard](#) to save filenames to a list as you encounter them on the screen. When viewing long lists of filenames, like the files available for download on your favorite BBS, you can select the files you want without having to copy the filename for each. A **Send** command sends all marked text to the terminal screen.

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.

## **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.

A number of file transfer protocols have been developed for personal computers, but the industry standard is **Zmodem**. This is the popular public domain streaming protocol.

Both Windows 95 and QmodemPro for Windows 95, as well as many BBSs and online services, use Zmodem as the default protocol. Unless you need to use a different protocol, Zmodem should work well for you. It is fast and easy.

Zmodem does not require you to send a command to it to start a download, it's automatic.

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.

If you need to use an alternative to Zmodem, other protocols are discussed in the QmodemPro for Windows 95 User Guide.

## **Batch Uploads**

An easy way to create a list of files for a batch upload is to use the Windows Explorer. The Explorer allows you to view the contents of drives and subdirectories, and when it is used with QmodemPro for Windows 95, you can drag and drop files from Explorer right into your Batch Upload window.

If you accidentally drop the files into the upper window, it will be copied to the directory shown in that window. To undo this mistake, click the right mouse button and select **Delete** to remove the file from the directory. You can also grab the files and drag them back, or, if you used Explorer to move the original files, you can choose **Edit/Undo** from the Explorer menu.

Note that the files in the QmodemPro for Windows 95 file list are sorted alphabetically, and also note that you can't add the same file more than once to the upload list.

## **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.

## **MD5**

If you have entered your password into the phonebook entry, the next time you connect to that system, you may not be asked for a password. The reason for this is MD5 (RSA Data Security, Inc.'s MD5 Message Digest Algorithm). This is an encryption system that reads your password from the phonebook entry, encrypts it, and sends it automatically to the remote system. On systems that support MD5, the remote system will unencrypt your password, so no password need actually be typed in. This affords a much higher level of security than typing in your password.

## Converting Your Phonebooks

QmodemPro for Windows 95 provides you with an automatic way to convert your phonebooks with the Conversion Wizard, but scripts are not compatible.

To convert the phonebook from the DOS versions of Qmodem, QmodemPro for DOS, QmodemPro for Windows, and Qmodem Test Drive, as well as Procomm Plus for Windows v 1.0 and v2.0, Procomm Plus DOS v1.1, Procomm Plus DOS v2.0, WinComm 1.1, Telix DOS v3.x, Telix for Windows v1.x, and Crosstalk, start QmodemPro for Windows 95 and open the phonebook or click the mouse on the [Dialer](#) button. Use the Phonebook menu command [File/Import](#) and select the name of the old phonebook. You will be guided by a “conversion wizard” through the conversion process. A wizard is a series of dialog boxes designed to take you step-by-step through a process.

The conversion wizard will help you to create a new phonebook using the same name with the extension .PHN. If using the same name would overwrite an existing file, you will be prompted to select a unique name. The QmodemPro for Windows 95 default dialing directory is C:\PROGRAM FILES\MUSTANG SOFTWARE\QMODEMPRO\QMWIN.PHN.

Due to changes in structures, DOS script files cannot be converted to the new Script Language Interface for Qmodem (SLIQ) format.

[Script](#) files used with QmodemPro for Windows can be recompiled automatically for use with QmodemPro for Windows 95. Copy the scripts into the new phonebook script directory. Enter the script name to the new phonebook entry. When you dial this number, QmodemPro for Windows 95 will automatically recompile the script using the Windows 95 format.

Some older Qmodem and QmodemPro for Windows scripts may not run. You will need to remove all fax, device, and direct-to-modem commands, then recompile.

## A note on conversions

Some of the features in QmodemPro for Windows 95 are completely new. QmodemPro for Windows 95 is the most versatile version yet.

You may be familiar with fax features in older versions of Qmodem. Because TAPI allows programs to "share" modems, Windows 95 includes a fax software package that can share resources with other applications. Now, you can send a fax with any information from QmodemPro for Windows 95 by using the **Send** command. Fax options in QmodemPro for Windows 95 have been removed.

Also, there are no devices in QmodemPro for Windows 95. Windows 95 takes care of all the device (now called modem) information. You are still able to fine tune your modem in QmodemPro for Windows 95, but in most cases, you won't need to.

If for some reason you must "talk" directly to your modem, you now must tell QmodemPro for Windows 95 and Windows 95. To issue commands directly to your modem, you must first select the [Direct Connection](#) option.

You may find that your scripts need to be adjusted and recompiled to remove device, fax, or modem commands that are no longer used by QmodemPro for Windows 95.

If you have used the DOS version of QmodemPro, Qmodem v5, Qmodem Test-Drive, or the older shareware version of Qmodem, you will find QmodemPro for Windows 95 is substantially different.

## The Main Menu

When you first open QmodemPro for Windows 95, you will see the main menu. This is where the menu commands, toolbar, macrobar, statusbar, and terminal screen are launched from.

Menu commands are accessed through this menu. Click on the menu command to execute it, or bring up a dialog box if more information is required. Toolbar commands can be accessed from the toolbar, which is completely configurable, as is the macrobar. Refer to your manual for more information about configuring these options.

The menu command line offers you options for [Files](#), [Edit](#), [View](#), [Connection](#), [Scripts](#), and [Tools](#). It also gives you access to help.

The [terminal screen](#) is where all of your online data will normally appear.

The macrobar appears at the bottom of the screen. You can have a different macrobar for each phonebook entry, if you want.

Under the macrobar is the statusbar. This shows you the current state of QmodemPro for Windows 95.



## Customizing the Desktop

You can customize the look of your desktop by using the [Tools/Options/Desktop](#) menu choice.

The desktop pattern is a simple repeating geometric pattern made of two color 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 [Color](#) options allow you to change the colors that make up the pattern. Use the foreground and background color buttons to view a list of available color options.

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.

If you have a sound card in your PC, QmodemPro for Windows 95 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 **Tools/Options** menu and select the [Sounds](#) property sheet to make changes. A property sheet will pop up showing a list of sound events you can modify. Highlight the entry you want to edit, and click the **Name/Browse** button. A standard Windows Common Dialog Box will pop up showing a list of WAV files in the bottom window, and options to search different drives and directories in the top window.

## 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.

[Top Ten Cool Things about QmodemPro for Windows 95](#)

[Order Information](#)

[Order Form](#)

## Wildcat! 4 BBS

Mustang Software unveiled the first Wildcat! BBS in 1986. Since its initial release, Wildcat! has emerged as the world's most popular Bulletin Board System (BBS) software with more than 30,000 Sysops worldwide. A centralized, PC-based information system, Wildcat! is packed with more intelligent features and value than any other BBS product available. Callers can send, receive and view messages and files, access external applications, and chat 24 hours a day. This full-featured software is available in four configurations for the hobbyist, small business and Fortune 1000 corporations, and allows simultaneous access from one to 250 users. Since Mustang offers complete BBS solutions, Sysops don't need to go searching for third-party add-on products.

## Wildcat! Single Line

Our economical and easily-installed flagship product requires only a telephone line, a personal computer, and a modem to begin posting messages and routing files around the world. Wildcat! 4 maintains its dominance in the BBS market by giving you greater flexibility and power. Intelligent features include:

- A .GIF thumbnailer that lets your callers download and preview a sample of multiple .GIF files in one combined bitmapped file;
- An industry first, spell-checking editor to allow your callers to audit their text before posting their messages;
- RIPscrip graphics to provide your callers with a graphical user interface (GUI) and allow them to navigate the BBS with their mouse;
- A free-form menuing system that gives you the ability to create a custom menu structure and command system;
- Support for over 32,000 message conferences and file areas to give your BBS plenty of room to grow;
- Sysop-defined language files that allow your callers to select the BBS language interface of their choice;
- wCDRAW BBS display screen editor that lets you customize the look of your BBS interface;
- The ability to perform all BBS Sysop functions from remote so you can maintain your data while you're on the road.
- Support for Serial, DigiBoard, FOSSIL and OS/2 com driver.

These product features are just a few of the reasons why Mustang Software maintains its leadership as the Number 1 BBS supplier worldwide. Our liberal upgrade policy allows Sysops to start with our Single Line version and trade it in later for full credit to any of our MultiLine products.

**Suggested list price: \$129**

## Wildcat! MultiLine 10

This MultiLine package extends the functionality of our base product by supporting up to 10 simultaneous users through your Local Area Network (LAN), multitasking environment or combination. The MultiLine 10 allows concurrent use of database utilities without interrupting your online user activity, and adds a powerful online chat conferencing system, with up to 8 high speed connections on a single 486 or Pentium PC.

**Suggested list price: \$249**

## Wildcat! MultiLine 48

Designed for corporate environments, Wildcat! MultiLine 48 extends MultiLine 10 features by supporting up to 48 concurrent users. Your LAN users have all of the E-Mail, messaging and file functions of dial-up users, so you can increase corporate productivity, in-house communications and customer access.

**Suggested list price: \$499**

## Wildcat! MultiLine Platinum

The same range of powerful features and performance of our smaller versions, with take-it-to-the max support for up to 250 concurrent connections. The premium package for large networks and mega-sized commercial bulletin board systems.

**Suggested list price: \$799**

## Wildcat! BBS Suite

Only Mustang could revolutionize the industry by producing the first "BBS Suite". The Wildcat! BBS Suite includes our top-of-the-line Wildcat! MultiLine Platinum BBS software with the wcPRO database utility, wcFAX on-demand server, wcGATE Internet/Novell MHS mail gateway, and wcCODE development tool kit to give you the ultimate BBS solution.

Wildcat! BBS Suite processes statistics and creates graphs on Wildcat!'s activities; allows your customers to receive product literature and troubleshooting tips via fax; attaches your message databases to the Internet and MHS-compatible E-Mail products; and gives you all the power to fully customize Wildcat!'s interface and create custom BBS applications, all in one economical yet powerful package. The BBS Suite is the ultimate tool for the professional Sysop -- you!

**Suggested list price: \$999**

### **An Exciting Array Of Add-On Packages.**

Our powerful collection of add-on utilities will enhance any *Wildcat! 4* BBS. Developed by the same Software Engineers that brought you *Wildcat! 4*, these utilities have the same easy-to-use interface as the rest of the *Wildcat! 4* family, and are fully supported by Mustang Software, giving you a single-source for all your BBS needs.

### **wcPRO Utilities For The In-Control Sysop**

wcPRO provides powerful user, message and file database maintenance capabilities. You can create custom reports with more than 20 pages of online statistics, create custom command macros for nightly batch processing, and export BBS data into other formats for external applications.

Our relational database handling makes managing your BBS callers, files and messages quick and easy. The wcPRO utilities package also includes our fax-on-demand processor, wcFAX, so your callers can request data online and receive product literature, registration forms, and price list documents via their fax.

**Suggested list price: \$99**

### **wcGATE Internet/MHS Messaging Gateway**

A sophisticated messaging gateway, wcGATE enables your callers to send and receive Internet E-Mail and news right from your BBS without having to know a single Internet command. Built-in support for satellite connections means your callers can have a constant flow of Internet information at their fingertips. wcGATE also allows you to transfer BBS messages to your company's favorite MHS-compatible E-Mail system. Outside sales-people and remote offices can leave messages on the BBS and have them directed to the appropriate mail box on your LAN using Novell's Message Handling Service.

**Suggested list price: \$149**

### **wcCODE Custom Online Development Engine**

wcCODE allows you to develop custom applications to enhance or change Wildcat's operation. BASIC-like syntax combined with an Integrated Development Environment makes it easy to use the more than 200 built-in functions, including access to *Wildcat!*'s internal operations. wcCODE allows you to customize *Wildcat!* and write applications which will work on any *Wildcat! 4* BBS. Easy-to-read manual and sample applications like our wcLIST BBS Lister and an on-line order system are included.

**Suggested list price: \$149**

### **wcBILLING Accounting & Billing Module for Wildcat!**

Turning a profit with your Wildcat! BBS just got a little bit easier. With wcBILLING, the newest addition to the Wildcat! product line, you now have a tool to implement a real-time billing system on your BBS with the flexibility to credit or debit your caller's account balances by the minute, for reading and writing messages, uploading and downloading files, using doors and chatting.

**Suggested list price: \$149**

### **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.

**Suggested list price: \$99**

### **Off-Line Xpress for Windows**

**NEW!** Windows version. 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. DOS version also available.

**Suggested list price: \$49**

### **Where can I buy these great products?**

#### **Retail Stores Selling MSI Products**

- Software Etc. For location nearest you call (800) 328-4646
- CompUSA For location nearest you call (800) 266-7872
- A-Plus Software (Canada) For location nearest you call (800) 265-6657
- Computer City For location nearest you call (800) THE CITY (843-2489)

The following retail stores also carry some or all Mustang Software products: Babbages, Best Buy, Egghead, Electronic Boutique, Elek-Tek, Fry's Electronics, Media Play, Software City, Waldensoftware, MicroCenter, Computerland, Fry's Electronics, Software Spectrum.

#### **Mail Order Companies Selling MSI Products**

- 800 Software (800) 888-4880
- SoftChoice — Canada (800) 268-7638
- BBS Technologies (209) 498-0200
- Micro WareHouse (800) 367-7080
- Programmers Provantage (800) 336-1166
- Windows Exchange (800) 845-1900
- NECX DIRECT - On the NET Live!

## **Top Ten Cool Features in QmodemPro for Windows 95 v2.0**

### **10. RIP Graphics terminal emulation support gives you a Point & Shoot GUI interface on thousands of BBSs around the world.**

RIP stands for Remote Imaging Protocol and it's the graphic terminal emulation standard supported by Wildcat! BBS, Major BBS, PC Board, TBBS and several other popular BBS software platforms. When you dial a RIP-capable BBS you can navigate with your mouse and enjoy the eye-popping graphics. It's the ultimate interface for the BBS surfer!

### **9. Powerful 32-bit, multithreaded design lets you take full advantage of Windows 95's multitasking.**

To benefit from the 32-bit horsepower provided by Windows 95 you need applications specifically designed to harness it. Using your old 16-bit communications software in a 32-bit environment will work, but it will adversely affect the performance of the entire system. QmodemPro delivers advanced features and performance in a specifically designed 32-bit package that keeps other Windows 95 applications humming.

### **8. MAPI links allow you to send session text, images and files to other MAPI-aware applications.**

It's easy to send any marked text, capture files or even the scrollback log to another user via any MAPI (Mail Application Programming Interface) compatible application like Exchange, MS Mail, cc:Mail or Notes.

### **7. Use QmodemPro as a full-featured Telnet client.**

Telnet into computers and BBSs around the world using all the features you're used to in a dial-up connection -- robust terminal emulations, speedy file transfers, and customized keyboard maps and macros. Plus, QmodemPro supports RSA Data Security, Inc. MD5 Message-Digest Algorithm for password security.

### **6. The most powerful and flexible phonebook you've ever seen!**

QmodemPro gives you a true Windows 95 phonebook. Each entry can have its own icon and you have four ways to view the phonebook (Large Icons, Small Icons, List and Details). The phonebook also supports up to 5 numbers per entry, stores a unique user ID and password for each entry, supports data, telnet and voice in the same phonebook and can be configured to load automatically when QmodemPro for Windows is started.

### **5. QuickLearn script recorder makes it super simple to automate repetitive actions such as logging onto a BBS or on-line service.**

Writing scripts has never been easier! All you need to do to have QmodemPro create a logon script is define a script name in the phonebook and dial the number. QmodemPro will automatically toggle into QuickLearn mode, record your logon and create a script for you to automate future logons.

### **4. The SLIQ compiler and debugger give you all the tools you need to develop custom applications in SLIQ (Script Language Interface for QmodemPro).**

SLIQ is similar to the popular BASIC programming language so it's easy to learn and use, yet powerful enough for the advanced user to develop specialized applications. QmodemPro's Host Mode script provides an example of the power and flexibility of SLIQ.

### **3. TAPI support means QmodemPro will share your modem with other TAPI-compliant applications.**

You can run QmodemPro concurrently with the Microsoft Exchange Fax system, Dial-up networking, Microsoft Network, or any other TAPI (Telephony Application Programming Interface) compatible application without having to worry about port conflicts.

### **2. OLE 2.0 Drag & Drop support lets you drag phonebook entries right to your desktop.**

You can easily create shortcuts with QmodemPro's Drag and Drop support. Commonly dialed numbers are then just a mouse-click away, even if you don't have QmodemPro currently running. You can also drag & drop files directly from the Windows Explorer into QmodemPro's file transfer window. Just mark the files you wish to upload in the Windows Explorer and just drag them to the Send Files window in QmodemPro. You can even mark the files before connecting to save time online.

### **1. GIF/BMP/JPEG Picture File Viewer (PFV) that lets you zoom any**

**graphic file up to 1600%, even while the file is still being downloaded!**

Just click the Zoom button on the toolbar and then left-click to zoom in and right-click to zoom out. You can even print or cut and paste the image to other applications. Way cool!

## Ordering Information for QmodemPro for Windows 95

Check out the [Top Ten Features](#) to find out why QmodemPro for Windows 95 v2.0 is *The Premier Communications Solution for Windows 95*.

QmodemPro is available direct from Mustang Software, Inc. for \$129.00 plus shipping and handling. You can order QmodemPro by calling toll-free +1 (800) 208-0615 or +1 (805) 873-2500, or order by fax or mail by printing out this handy [Order Form](#).

You can also purchase QmodemPro for Windows 95 v2.0 at your local CompUSA, Computer City, Software Etc, Babbages, Egghead Software, Incredible Universe, Best Buy, Office Max, Office Depot, Electronics Boutique, Elek-Tek, Fry's Electronics and other fine software stores everywhere.

When you purchase QmodemPro you will receive:

- Comprehensive 300+ page User Guide and 325+ page Script Guide
- Registered user access to the MSI HQ BBS
- Free technical support by phone, MSI HQ BBS, Compuserve, America Online and Genie
- Free subscription to the MSI Monitor Newsletter
- Special discounts and offers on selected MSI products
- Plus much, much more!

## QmodemPro for Windows 95 v2.0 Order Form

**Yes, I want The Premier Communications Solution For Windows 95!** Send me QmodemPro for Windows 95 v2.0 for only \$129 plus shipping.

Please print your name and shipping address below (No PO Boxes, please):

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

City: \_\_\_\_\_ State/Province: \_\_\_\_\_

Zip/Postal Code: \_\_\_\_\_ Country: \_\_\_\_\_

Daytime Phone number: \_\_\_\_\_

### Payment method:

\_\_\_\_\_ Check/Money Order (US Funds Only, Please)

\_\_\_\_\_ Credit Card

Visa, MC, Discover \_ \_ \_ \_ \_ -- \_ \_ \_ \_ \_ -- \_ \_ \_ \_ \_ -- \_ \_ \_ \_ \_

American Express \_ \_ \_ \_ \_ -- \_ \_ \_ \_ \_ -- \_ \_ \_ \_ \_ -- \_ \_ \_ \_ \_

Signature: \_\_\_\_\_ Expiration Date: \_\_\_\_\_

Note: Be sure to include shipping and applicable taxes. Shipping for US customers is \$10. International customers contact Mustang Software, Inc. (800-208-0615 or 805-873-2500) for shipping options and costs. California residents must include local sales tax.

### To Order:

#### By Phone:

Please have your credit card and order ready.

Inside US and Canada, call Toll Free: (800) 208-0615  
Outside US: (805) 873-2500

#### To Fax Your Order

Our fax number is (805) 873-2599

If you're paying by credit card, you may also fax this entire form if you wish. Orders faxed on weekends and holidays are usually processed the next business day.

#### To Mail Your Order

Send your check, money order, or credit card information with this form to:

Mustang Software, Inc.  
PO Box 2264  
Bakersfield, CA 93303-2264

Q95 Demo 11/3/95

{button Print Order Form,Print() }



## Converting Your Phonebooks

QmodemPro for Windows 95 provides you with an automatic way to convert your phonebooks with the Conversion Wizard, but scripts are not compatible.

To convert the phonebook from the DOS versions of Qmodem, QmodemPro for DOS, QmodemPro for Windows, and Qmodem Test Drive, as well as Procomm Plus for Windows v 1.0 and v2.0, Procomm Plus DOS v1.1, Procomm Plus DOS v2.0, WinComm 1.1, Telix DOS v3.x, Telix for Windows v1.x, and Crosstalk, start QmodemPro for Windows 95 and open the phonebook or click the mouse on the [Dialer](#) button. Use the Phonebook menu command [File/Import](#) and select the name of the old phonebook. You will be guided by a “conversion wizard” through the conversion process. A wizard is a series of dialog boxes designed to take you step-by-step through a process.

The conversion wizard will help you to create a new phonebook using the same name with the extension .PHN. If using the same name would overwrite an existing file, you will be prompted to select a unique name. The QmodemPro for Windows 95 default dialing directory is C:\PROGRAM FILES\MUSTANG SOFTWARE\QMODEMPRO\QMWIN.PHN.

Due to changes in structures, DOS script files cannot be converted to the new Script Language Interface for Qmodem (SLIQ) format.

[Script](#) files used with QmodemPro for Windows can be recompiled automatically for use with QmodemPro for Windows 95. Copy the scripts into the new phonebook script directory. Enter the script name to the new phonebook entry. When you dial this number, QmodemPro for Windows 95 will automatically recompile the script using the Windows 95 format.

Some older Qmodem and QmodemPro for Windows scripts may not run. You will need to remove all fax, device, and direct-to-modem commands, then recompile.

## Macro System Functions

Macros can be used to perform many functions automatically. These can be assigned to buttons located at the bottom of the terminal window. Click on the macro you want to know more about:

[@ADDLFTOCR \[on | off\]](#)

[@AUTOANSWER \[on | off\]](#)

[@BREAK](#)

[@CAPTURE \[filename | off\]](#)

[@CLS](#)

[@DIAL \[ENTRY | GROUP | SEARCH | MANUAL\] string](#)

[@DIALMANUAL\[NNN-NNNN,AAA,CCC\]](#)

[@DIRECTCONNECT](#)

[@DOORWAY \[on | off\]](#)

[@DOWNLOAD \[filename/dir \[protocol\]\]](#)

[@DUPLEX \[on | off\]](#)

[@EDIT \[filename\]](#)

[@EMULATION \[emulation\]](#)

[@EXIT](#)

[@EXPLOREDOWNLOAD](#)

[@EXPLOREUPLOAD](#)

[@HANGUP](#)

[@HELP](#)

[@HOST](#)

[@MANUALDIAL\[NNN-NNNN,AAA,CCC\]](#)

[@MODEMPROPERTIES](#)

[@PASSWORD](#)

[@PHONEBOOK \[phonebookname\]](#)

[@PRINTER \[on | off\]](#)

[@QUICKLEARN \[scriptname\]](#)

[@RESETEMULATION](#)

[@SCRIPT \[scriptname\]](#)

[@SCRIPTCOMPILE \[scriptname\]](#)

[@SCRIPTDEBUG \[scriptname\]](#)

[@SCRIPTEDIT \[filename\]](#)

[@SCROLLBACK \[on | off\]](#)

[@SCROLLBACKRECORD \[on | off\]](#)

[@SHELL \[filename\]](#)

[@SPLITSCREEN \[on | off\]](#)

[@STAMP \[string\]](#)

[@STRIPHIBIT \[on | off\]](#)

@TRAPSCREEN [filename]

@UPLOAD [filenames [protocol]]

@USERID

@VIEW [filename]

@VIEWPICTURE [filenames]

@XONXOFF [on | off]

**@ADDLFTOCR [on | off]**

This is the same as menu command Connection/Add LF to CR.

**@AUTOANSWER [on|off]**

This is the same command as Auto Answer on/off.

## **@BREAK**

This is the same as menu command Connection/Break.

**@CAPTURE [filename | off]**

Toggles the capture file on and off.

**@CLS**

Clears the terminal screen.



**@DIAL [entry|group|search|manual] string**

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

**@DIALMANUAL [NNN-NNNN,AAA,CCC]**

This is the same as the menu command Dial/Manual Dial. N is the phone number, A is the area code, C is the country code. IF you do not specify an area or country code, the dialer will use your default.

## **@DIRECTCONNECT**

Opens a direct connection, bypassing TAPI. Any valid COM port may be specified. If no parameter is specified, a direct connection dialog box is opened. This is the same as the menu command Connection/Direct Connect.

**@XONXOFF [on|off]**

Changes XON/XOFF flow control to specified parameter. Same as menu command Phonebook/Edit Phonebook Entry/Connect Using/ Advanced/Xon/Xoff.

**@VIEWPICTURE [filenames]**

Opens the Picture Viewer and loads in specified filenames.

**@VIEW [filename]**

Opens the text editor and loads in the specified file.

**@USERID**

Extracts the user ID from the USERID field in the current phonebook entry. This is valid only when dialed from the phonebook.

**@UPLOAD [filenames [protocol]]**

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



**@TRAPSCREEN [filename]**

Saves the current screen to the specified filename. This is the same command as menu command Edit/Screen to File.

**@DOORWAY [on|off]**

Same as menu command Connection/Doorway. This command can only be used if the remote system supports it.

**@DOWNLOAD [filename/dir [protocol]]**

Downloads the specified file, using the specified protocol.

**@DUPLEX [on | off]**

Turns full duplex on or off. Same as menu command Connection/Full Duplex.

**@EDIT [filename]**

Opens the Editor and loads the requested file.

**@EMULATION [emulation]**

Sets the emulation as specified. Same as menu command Connection/Emulation.

**@EXIT**

Exits QmodemPro for Windows 95 completely. This command cannot be used to close the phonebook, editor, or picture viewer.

**@EXPLOREDOWNLOAD**

Opens the Windows Explorer in your download directory.



**@EXPLOREUPLOAD**

Opens the Windows Explorer in your upload directory

**@HANGUP**

Sends a hangup command. Same as menu command Connection/Hangup.

**@HELP**

Opens the QmodemPro for Windows 95 Help System in the terminal window.

**@STRIPHIBIT [on |off]**

Turns the 8th bit strip feature on or off. Same as menu command Connection/8th bit strip.

**@STAMP [string]**

Records an entry into the log file.

**@SPLITSCREEN [on | off]**

Turns the split screen terminal view on or off. Same as menu command View/Split Screen.

**@HOST**

Executes the Host Mode script

**@MANUALDIAL [NNN-NNN,AAA,CCC]**

Dials a number manually. N is number, A is area code, C is country code. If no parameters are set, the Manual Dialog Box is opened.



## **@SHELL**

Go to a Command Line prompt and execute the requested application.

**@SCROLLBACKRECORD [on | off]**

Turns the scrollback record option on or off. Same as the menu command View/Scrollback Record.

**@MODEMPROPERTIES**

Opens a property sheet allowing the configuration of modem properties. This is the same as the menu command Phonebook/Edit Phonebook Entry/Connect Using/Configure.

**@SCROLLBACK [on | off]**

Opens the scrollbar log for viewing or closes it.

**@PASSWORD**

Extracts the Password field from the current phonebook entry. This only works when the number is dialed from the phonebook.

**@SCRIPTEDIT [filename]**

Opens the scripts editor and loads the requested script.

**@PHONEBOOK [phonebookname]**

Opens the requested phonebook. Same as menu command Phonebook/File/Open.

**@SCRIPTDEBUG [scriptname]**

Opens the script debugger and loads in the requested file.



**@PRINTER [on | off]**

Toggles the print capture option on or off. Same as menu command File/Printer Toggle.

**@QUICKLEARN [scriptname]**

Executes the QuickLearn script and saves it with the specified filename.

**@RESETEMULATION**

Resets the current emulation to its default configuration.

**@SCRIPT [scriptname]**

Executes the requested script. Same as menu command Scripts/Execute.

**@SCRIPTCOMPILE [scriptname]**

Compiles the requested script. Same as menu command Scripts/Compile.

## New in QmodemPro for Windows 95

- Put a shortcut to QmodemPro for Windows 95 on your desktop. A **shortcut** can be created to QmodemPro for Windows 95 and placed on the desktop.
- Send mail or compose a letter with text or graphics from your terminal window. [MAPI](#) (messaging application program interface) support allows you to send text directly from the terminal window to mail applications (like Microsoft Exchange).
- A [file clipboard](#) has been added, allowing you to drag filenames from the terminal window directly to a download list. You can save groups of filenames, ready for download. You will never need to manually copy a filename again.
- Put a favorite phonebook entry right on your desktop. The [OLE 2](#) “**drag and drop**” feature is supported throughout QmodemPro for Windows 95 in the phonebook, terminal window, editor, and File Clipboard.
- With fully multithreaded protocols, even when your PC is very busy, the protocol will be able to continue.
- QmodemPro for Windows 95 allows multiple translation tables to be defined and a different one selected for each phonebook entry.
- QmodemPro for Windows 95 supports international dialing and credit card billing numbers automatically, using Windows 95’s dialing features.
- Added [MD5](#) (RSA Data Security, Inc.’s MD5 Message Digest Algorithm). This feature lets you send encrypted passwords for greater security when logging on to Internet sites, Telnet sites and BBSs supporting MD5.
- When making voice calls, a dialog box has been added that shows you the length of your call, allows you to hang up, and logs all the information to a log file.

## **Contents**

The following Help Topics are available:

[Launching the Editor](#)

[Keyboard Commands](#)

[Marking Text](#)

[Moving Text](#)

[Copying Text](#)

[Deleting Text](#)

[Undo/Redo](#)

[Finding Text](#)

[The Script Editor](#)

[Printing and Sending Documents](#)

[The Scripts Editor](#)

[Changing the Look of the Editor](#)

[The File Menu](#)

[The Edit Menu](#)

[The View Menu](#)

[The Compile Menu](#)

[Scripts and SLIQ](#)

## **The Editor**

The QmodemPro for Windows 95 editor can be used to edit scripts or text files. If you have used other word processors and text editors in Windows, you should be familiar with most commands, icons and shortcut keys used by the *QmodemPro for Windows 95* Editor.

[Launching the Editor](#)

[Keyboard Commands](#)

[Marking Text](#)

[Moving Text](#)

[Copying Text](#)

[Deleting Text](#)

[Undo/Redo](#)

[Finding Text](#)

[The Script Editor](#)

[Changing the Look of the Editor](#)

[Printing Text](#)

[Scripts and SLIQ](#)



## **The Script Editor**

If you choose the **Scripts/Edit** command from the *QmodemPro for Windows 95* main menu to open the Qmodem Editor, the Editor will open with special features used especially for creating and editing scripts.

You can also launch the script editor by using the Start/Run command from the taskbar and typing:

```
C:\QMWIN\QMEDITOR.EXE /SCRIPT
```

The primary differences between the two versions of the Editor deal with scripting manipulation. The Script Editor has a built-in Compiler feature, to allow quick compiling while you are editing.

It also offers a Syntax Highlight option. This option allows you to show specific types of script language (syntax) in user-defined colors. This allows you to visually differentiate syntax at a glance. For a detailed list of these features, refer to Book 2 of the *QmodemPro for Windows 95* documentation, the SLIQ guide.

## Keyboard Shortcuts

CTRL+O	Open an existing file
CTRL+S	Save the current file
CTRL+P	Print the current file
CTRL+N	Create new file
CTRL+T	Delete word right
CTRL+V	Paste from Windows clipboard
CTRL+X	Cut marked text to Windows clipboard
CTRL+Y	Delete line
CTRL+C	Cut to Windows clipboard
CTRL+H	Paste from Windows clipboard
CTRL+Z	Undo editing
CTRL+F	Find text
CTRL+HOME	Top of screen
CTRL+END	End of file
F3/Shift-F3	Search Again (forward/backward).

## Launching the Editor

You can launch the Qmodem Editor from a *QmodemPro for Windows 95* toolbar icon, if you have configured the toolbar to include this icon. To add the Editor icon, click **Tools/Customize** and select **Toolbar**, or click the right mouse button on the toolbar and select **Properties**. Highlight the Editor option in the Available Buttons section, and double click or click Add to add it to **Toolbar Buttons**. The icon will be added to the toolbar.

You can also open the Editor from a command line without even opening *QmodemPro for Windows 95* . From a command line (choose Start/Run from the Windows Taskbar), type

```
C:\QMWIN\QMEDITOR.EXE
```

or

```
C:\QMWIN\QMEDITOR.EXE /SCRIPT
```

to launch the Script Editor. If you have installed *QmodemPro for Windows 95* on a different drive, change the command line to reflect the appropriate drive and directory for your particular setup.

**Marking Text**

You can mark text with the mouse. After the text is marked, press CTRL and hold down the left mouse button. Move the cursor to the place where you want to copy to, and release the mouse button.

Another alternative is to mark the text with the mouse and then click the right mouse button on the highlighted text.

## **Moving Text**

First, mark the text you want to move. Then cut the marked text to the Windows 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 copy the selected text. Using your mouse, select the menu command **Edit/Paste**. Or, press CTRL+V.

Another alternative is to mark the text with the mouse and then drag the text to the new position.

## **Copying Text**

First, mark the text you want to copy. Then copy the marked text to the Windows Clipboard using your mouse to select the menu command **Edit/Copy**. Or, press CTRL+C.

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+V.

Another alternative is to mark the text with the mouse and then click the right mouse button on the highlighted text. Select **Copy** from the drop down box options.

## **Deleting Text**

Use the backspace or DEL key when you want to delete text from the current document but you have text on the Windows Clipboard that you want to keep.

If you want to cut text but save it to the Windows Clipboard, highlight the text you want to cut, then use the Cut icon (it looks like little scissors) or CTRL+X

Another way to delete text is to mark the text with the mouse and then click the right mouse button on the highlighted text. Select **Delete** from the drop down box options.

To delete one character at a time, press backspace 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 backspace or DEL. To place the text onto the Windows Clipboard, choose **Cut** from the **Edit** menu.

## **Undo/Redo**

You can **redo** changes, and restore your text to the way it was before you undid the last command, with the **Edit/Undo** command or CTRL+A.

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 CTRL+Z.

## 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 the place you want the search to begin. From the Edit menu, choose **Find**. Type in the text you want to find.

If you want to match case exactly, select the **Case Sensitive** option.

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

To find the next occurrence of the text, choose the **Find Next** button again.

You can also use the **Find** icon on the toolbar to search for text. Click the icon (it looks like a magnifying glass) and a **Find** dialog box will appear.

You can replace specific text by using the **Replace** command. Much the same way that **Find** and **Find Next** find text, **Replace** will allow you to replace specific text. Enter the text you want to get rid of, and the text you want to replace it with. If you want to match capitalization exactly, select the **Match Case** check box. Click on **Find Next**, verify that you want the text replaced, and click **Replace**.

You can use the **Edit/Replace All** command to globally replace text in your file.

F3/Shift-F3 allows you to use the Search Again (forward/backward) command.



## **Print/Send**

You can print your file by selecting **File/Print**, or pressing CTRL+P. An options box will appear, allowing you to select the printer to use, page range, the number of copies, and print setup.

The Qmodem Editor has a built in Print Preview feature. Before printing a text file, you can click on File/Print Preview to see a single or multiple page preview of how the document will look before you print it.

Files open in the Editor can be sent directly as a mail message, appended to a mail message, or as a fax directly from the Editor. Selecting the **File/Send** menu command from the Editor main menu will launch Microsoft Exchange, allowing your file to be included as all or part of any Exchange application.

## Changing the Look of the Editor

Use the Editor's View command to change the look of the Editor. You can turn off the toolbar or status bar with the **View/Toolbar** or **View Status bar** command. Choose Toolbar to toggle the toolbar on and off. If the toolbar is visible, a checkmark will appear beside it when you use this command. Toggling the status bar on and off is done in exactly the same way as the toolbar, using the **View/Status bar** command.

The **View/Options/Fonts** command will bring up a property sheet with options for setting the default fonts for the edit window. **View/Options/Editor** lets you set tab definitions, word wrap, and auto indent options.

## **The View Menu**

The View menu allows you to change the look of the Editor.

**Toolbar** Toggles the Toolbar Visible command on and off. When **on**, the toolbar is visible from the editor window.

**Statusbar** Toggles the Statusbar Visible command on and off. When **on**, the statusbar is visible from the editor window.

**Options** Allows you to set fonts, tabs, word wrap, and auto-indent options.

## **The Edit Menu**

The *QmodemPro for Windows 95* edit menu allows you to manipulate data inside files, with common editing commands. Here is a list of options, and a brief description of their function available in the Edit menu.

**Undo** Undoes (reverses) the last action you performed.

**Redo** Restores the document to the way it was before you performed an undo. "Undoes" the last "Undo".

**Cut** Deletes the marked selection from your document and places it into the Windows clipboard.

**Copy** Copies the marked selection of your document to the Windows clipboard without deleting it from your document.

**Paste** Pastes the contents of the Windows Clipboard to the current cursor position.

**Delete** Deletes the marked selection from your document.

**Find** Opens a Find dialog box, allowing you to specify text to find in the currently active document.

**Replace** Works in conjunction with the Find command. Allows you to globally search and replace text within your document.

**Goto** Places the cursor at a specified line number in the currently active document.

**Find Error** (Script Editor Only) Finds and moves the cursor to a run time error in the source code.

## **The File Menu**

The *QmodemPro for Windows 95* Editor file menu has options allowing you to open, save, and print files. Here is a list of options, and a brief description of their function in the File menu.

**New** Opens a new file. The CTRL+N keys can also be used to open a new file.

**Open** Opens the Windows common dialog box, allowing you to choose the file you want to open from available drives and directories. The CTRL+O keys will also open this dialog box.

**Save** Saves the currently active file, including any changes you have made. Overwrites the original file. You can also use the key command CTRL+S to save a file. If you want to save changes without overwriting the original, use Save As. Because of Microsoft's Win95 requirements, the Qmodem Editor file extension is now '.QED' (Qmodem Editor File) instead of '.TXT'.

**Save As** Opens a Save As dialog box, allowing you to save changes to your document by saving it with another filename. This allows you to keep your original document intact.

**Print Preview** Shows a representation of the entire document as it will look when printed.

**Print Setup** Opens the Windows Print Setup dialog box, allowing you to change the setup of your printer, including which printer to use, the printer properties, paper orientation, and paper source.

**Send** Allows you to send the document directly to an E-mail message. Opens Microsoft Exchange or your specified electronic messaging system.

## **The Compile Menu**

The Script Editor offers an extra command menu, the Compile menu. This offers features specific to editing scripts. The features on the Compile menu are:

**Compile** Compiles source code active in the Editor window.

**Primary File** Defines the focus file for the compile. This defines the primary file when more than one file is being used. When a primary file has been defined, the filename will appear next to this command in the menu.

**Clear Primary File** Clears the primary file, allowing you to define a new file as primary.

## Scripts and SLIQ

What is a script? A script program is a set of instructions that you give *QmodemPro for Windows 95* that tells it how to perform a particular task. Its structure and syntax is similar in many ways to the popular Basic programming language, but you don't need to be a programmer to make use of Script Language Interface for *QmodemPro for Windows 95* (SLIQ).

Depending on the complexity of its construction, a SLIQ script can be thought of as a command shortcut, a macro or a computer program. Scripts are an ideal way to automate repetitive tasks, but can also be used to create custom options for the flow of your telecommunications activity. A SLIQ script can make use of *QmodemPro for Windows 95* program functions as well as internal functions within Windows 95 itself.

SLIQ scripts are composed of lines of text that represent the activities to be performed. It's a compiled script language, meaning that the text representation of the script is converted to a binary file when it is complete. Compiling scripts not only allows them to execute faster, but lets you distribute executable copies of your scripts without the underlying text source code.

For the large majority of *QmodemPro for Windows 95* users, script usage will be limited to grouping a sequence of tasks into a single command, usually by using the QuickLearn feature. QuickLearn provides an interface to SLIQ that is as easy to use as a tape recorder; you just turn it on and it records your task for playback at a later time.

What can I do with a script?

SLIQ provides the means of performing almost any telecommunications [activity](#) desired.

To learn more about scripts and SLIQ, select a topic below. Refer to the SLIQ Script Guide (volume 2 in your documentation) for a more complete explanation of the SLIQ language.

[Design Elements of a Script](#)

[QuickLearn](#)

[Compiling a Script](#)

[Debugging a Script](#)

[Running a Script](#)

[Stopping a Script](#)

[General Information](#)

## **What can I do with a script?**

### Automated logon

While the steps to connect to various on-line services, host computers and BBS systems are very straightforward in *QmodemPro for Windows 95*, each host has its own procedures for getting your name, password and other information. Automation of this process is probably the most common application for a script program. It is also the one that lends itself best to the QuickLearn feature since it is repetitive in nature.

### File transfers

Automating the process of downloading files from a remote system is another application where scripts are handy. Although the QuickLearn feature can also be used for this type of activity, it may require modification to control the desired download directories or other download parameters.

### Fully automated sessions

Combining the two previous examples you can see that a more complex script is capable of logging on to a remote system, downloading files and perhaps additional activity such as retrieving new messages. Any activity that can be performed manually with *QmodemPro for Windows 95* can be automated using SLIQ.

### Unattended QmodemPro for Windows 95 operations

Perhaps the best example of an unattended script is the *QmodemPro for Windows 95* host mode. The functionality of host mode is discussed in the User Guide, but the actual operation of host mode is driven by a series of pre-written scripts that are included with *QmodemPro for Windows 95*. These scripts and several others will be used as examples as we explore the power of SLIQ.



## Design elements

SLIQ is patterned after the BASIC programming language. This decision is based on the desire to incorporate a number of advanced programming structures and concepts, while appealing to the largest audience. If you are familiar with BASIC you will be accustomed to many aspects of SLIQ. There will be some new commands specific to *QmodemPro for Windows 95* but you should feel very comfortable with writing and editing scripts.

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.

Scripts are stored in the directory specified for script files which is selected with the menu choice **Tools/Options/Paths**. SLIQ files use standard Windows 95 filenames with two different extensions. The source script files created with your editor uses the extension **.QSC** and when they are compiled into an executable form by SLIQ they are given the extension **.QSX**.

Script files are created and modified using any standard ASCII text editor. *QmodemPro for Windows 95* includes an internal editor that you access from the menu choices **Scripts/Edit** or **Tools/Editor**. The internal editor supports auto-indent and tabs for structuring your script source files in a readable format, however you are free to use an external editor or programming environment if desired.

## How are scripts created?

### The QuickLearn feature

The *QmodemPro for Windows 95* script language can be used for many applications without the need to learn anything about scripting or SLIQ. QuickLearn lets you to record keystrokes for playback at a later time and is a simple process.

The first way of starting a QuickLearn script is to revise a Phonebook entry and place the name of a new script in the Scripts field. The next time you dial this entry, *QmodemPro for Windows 95* will automatically record a script file.

The second way to create a QuickLearn script is to use the pull down menu choice **Scripts/QuickLearn** to begin recording a script.

When activated, *QmodemPro for Windows 95* displays the standard file selection dialog and request a name for the script to be created. All scripts should be saved in the directory designated for scripts which by default is C:\QMWIN\SCRIPTS. If your configuration is different you should save your scripts in the correct script directory.

Any valid Windows 95 filename may be used and the extension must be **.QSC**. This extension is automatically added when script names are specified in the script filename dialog box.

Once a filename is selected you are returned to terminal mode and every keystroke is recorded until QuickLearn mode is terminated by again selecting the **Scripts/QuickLearn** menu choice. While recording is active the terminal window status line displays the name of the script preceded by the "greater than" symbol (>) indicating that recording is active. In addition, a check mark is placed in front of the QuickLearn menu option.

Scripts created using QuickLearn can be viewed and edited just as any other script. The menu choice **Scripts/Edit** can be used to take a look at the scripts created with SLIQ's QuickLearn feature.

While QuickLearn is an extremely convenient facility, it has its limitations. QuickLearn can record your interactions with a remote computer and replicate them exactly, but it cannot add decision-making logic to the scripts it creates, or account for unexpected or changing conditions. When you run out of QuickLearn power, it's time to write or edit your own scripts.

### Script programming

Writing a new script from the ground up is a necessity in some cases. Any standard ASCII text editor can be used to create or modify a SLIQ script, including the *QmodemPro for Windows 95* internal editor.

To create a new script select the menu choice **Scripts/Edit** to display the standard file selection dialog box. The default directory in the dialog will display files in your scripts directory as defined in **Tools/Options/Paths**, and the default extension will be **.QSC**. Create a new script file by entering the name of the script to be created. It is not necessary to add the script extension **.QSC** since it is added automatically when the **Scripts/Edit** menu command is used to create a script file. Note that the alternate method of invoking the editor, using the menu choice **Tools/Editor/Edit a File** does *not* automatically add the extension **.QSC**.

To edit a SLIQ script use the same procedure as creating a new script, but select the desired script from the dialog box.

## Compiling a script

SLIQ is a compiled language. This means that the text of every **.QSC** text file is converted to executable code before being used by *QmodemPro for Windows 95*. Script source files (those ending in **.QSC**) are compiled either automatically or manually.

After creating a SLIQ script file you can manually perform a compile by selecting the **Scripts/Compile** menu choice. The compile process will read your script, check each line for proper syntax and create an executable **.QSX** file with the same name as the source file. If any errors are encountered the compile process will stop with a compile error message and invoke the editor. The editor cursor will be located as near the error as possible, enabling you to make the necessary changes. A complete list of compiler error messages can be found in the sections **Compiler Errors and Script Debugging**, later in this manual.

If a script has been created or modified but not yet compiled when a request to execute is received, the compile process is automatically invoked. The compile process may take a few seconds to several minutes depending on the size and complexity of the script.

## **Debugging a script**

Regardless of the simplicity of a script or the capabilities of the programmer, scripts are seldom written perfectly on the first attempt. Many times an error is simply a typographical mistake or improper syntax for a command. In these cases the errors are caught during the compile cycle and the editor cursor is placed at the approximate location of the error in the script source. This type of error is usually relatively easy to correct.

The harder type of error to locate and correct involves a script that follows correct syntax and compile guidelines, but fails to produce the desired results. In this case the script debugger is an invaluable tool. The debugger allows you to execute a script line-by-line while viewing the results. It can skip sections of code and even display the changing value of variables as each line is executed.

The operation of the debugger is discussed in its own section after the SLIQ language documentation. See the **Compiler Errors** section under **Debugging** for details.

## Running a script

Scripts can be executed in several different ways:

1. Scripts can be started with the menu command **Scripts/Execute**. When this option is used, you can select either the **.QSC** or **.QSX** filename for the desired script if both exist. If only the **.QSC** source filename exists, the script will be compiled to a **.QSX** file before execution.
1. Scripts can be executed by clicking on the toolbar button labeled **Execute Script**. This action displays the same dialog box that is called when the **Scripts/Execute** command is selected.
1. Scripts can be started automatically every time a Phonebook entry is dialed. A script name is associated with a specific dialing directory entry in the *Script* field for entries in the *QmodemPro for Windows 95* Phonebook. When connection is made, the script is executed automatically. These types of scripts are often referred to as *linked scripts* because they're linked to phonebook entries. The same script can be tied to multiple Phonebook entries, and can be used to take control immediately after a connection to automate repetitive logon processes. Linked scripts that automate logon activities are also called *logon scripts*.
1. Scripts can be started by clicking a macro key on the macro bar at the bottom of the screen that has been assigned to a script name using the @SCRIPT macro command. Once started, macro-linked scripts operate the same as a script executed from the pull-down menu.
1. Scripts can be executed immediately when *QmodemPro for Windows 95* is started. The command line syntax for launching *QmodemPro for Windows 95* with a script is:
  1. QMWIN FILENAME.QSX
  1. or
  1. QMWIN FILENAME.QSC
  1. which will cause a compile if needed.
1. Once initialization is complete, *QmodemPro for Windows 95* will load and execute the script specified on the command line.

Script extensions .QSX and .QSC were automatically registered with Windows 95 when *QmodemPro for Windows 95* was installed. This association allows you to double-click on the script file name within Explorer to launch *QmodemPro for Windows 95* with the selected script as a command line argument.

Regardless of how a script is invoked, it operates in the same manner. The Script icon on the toolbar is displayed in a depressed position during script execution and the terminal windows status line displays the name of the script being run.

## Stopping a script

You can stop a script at any time in several ways. The easiest is to click on the **Script** icon on the toolbar. A confirmation prompt dialog box is displayed before execution is terminated.

You can also toggle the menu selection **Scripts/Execute** to stop a script. This menu item displays a check mark during execution, which is removed after confirming that you want to stop the script.(New topic text goes here.)

## 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 listed with examples in the SLIQ Script Guide.

See also [Constants](#), [Types](#), [Variables](#), [Expressions](#), [Statements](#), [Multiple line IF statements](#), [Loop statements](#), and [Subroutines and Functions](#).

## Tokens

Tokens are the smallest meaningful units of text in a *QmodemPro for Windows 95* 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<sup>-2</sup>.

## 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 byval call captionstring case catch close const  
declare dial dialog dialogbox dim div 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 consist of text the compiler should ignore, but are useful for annotating your SLIQ programs. Comments are preceded by either the rem statement, a double forward slash (//) 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 can't 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
short
integer (or long)
real
string
```

### Simple types

There are five simple types whose characteristics are shown below:

<b>Type</b>	<b>Range</b>	<b>Format</b>
byte	0 to 255	8 bit unsigned
short	-32768 to 32767	16 bit signed
integer (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 person's 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 95 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 bob's 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:

<u>Operator</u>	<u>Notes</u>
$\wedge$	exponentiation
$*$ / \ mod	multiplication, division, integer division or same as, modulus
$+$ $-$	addition, subtraction
$=$ $<>$ $<$ $<=$ $>$ $>=$	equality and inequality operators
and	bitwise logical and
or	bitwise logical or
xor eqv	exclusive or, equivalence ( $(x \text{ eqv } y) = \text{not } (x \text{ xor } y)$ )
imp	implication, ( $(x \text{ imp } y) = ((\text{not } a) \text{ or } b)$ )

Parentheses are used to modify the above precedence order. Expressions within parentheses are always evaluated starting with the innermost set of parentheses. For example,

<u>Expression</u>	<u>Notes</u>
$3 + 4 * 5 = 23$	(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 don't.

Simple statements stand by themselves and don't 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" clause, 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 statements

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)
```

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. There are two exceptions to this rule. The first is when 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
```

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.

The second exception to the pass-by reference rule occurs when the BYVAL keyword is used. Refer to the BYVAL keyword for information and examples.

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.

## Script Command Reference

The following is a complete description of all available *QmodemPro for Windows 95* script commands. Script commands are presented in the following format:

### Script Command

A brief description of the command.

Syntax

A description of the command's syntax.

### Remarks

A more detailed description of the command, including parameters and their meaning, and how the command interacts with the rest of the script and the system in general.

### See also

Related script commands.

### Example

Most examples will be a complete script program that you can type in and run.

Click on the command you are interested in.

[\\$INCLUDE](#)

[ABS](#)

[ACTIVATE](#)

[ADDLFTOCR](#)

[ADDPHONEENTRY](#)

[AND](#)

[ASC](#)

[ATN](#)

[AUTOANSWER](#)

[BEEP](#)

[BREAK](#)

[BYTE](#)

[BYVAL](#)

[CALL](#)

[CAPTURE](#)

[CARRIER](#)

[CASE](#)

[CATCH](#)

[CHAIN](#)

[CHDIR](#)

[CHDRIVE](#)

[CHR](#)

[CLOSE](#)

[CLOSEPORT](#)

[CLS](#)

[COLOR](#)

[CONFIGCAPTUREFILE](#)

[CONFIGDOWNLOADPATH](#)

[CONFIGLOGFILE](#)

[CONFIGSCRIPTPATH](#)

[CONFIGSCROLLBACKFILE](#)

[CONFIGTRAPFILE](#)

[CONFIGUPLOADPATH](#)

[CONST](#)

[COPYFILE](#)

[COS](#)

[CSRLIN](#)

CURDIR  
CURDRIVE  
DATE  
DATEIMEDIFF  
DECLARE  
DEL  
DELAY  
DIAL  
DIALNEXT  
DIALOG  
DIALOGBOX  
DIM  
DO\_LOOP  
DOORWAY  
DOWNLOAD  
DUPLEX  
EDITFILE  
ELSE  
ELSEIF  
EMULATION  
END  
ENVIRON  
EOF  
EQV  
ERROR  
EXISTS  
EXIT  
EXP  
FINDFIRST  
FINDNEXT  
FIX  
FLUSH  
FOR\_NEXT  
FORMATDATE  
FORMATTIME  
FREEFILE  
FUNCTION  
GET  
GETCURRENTDATETIME  
GETFIRSTCOUNTRY  
GETMODEMCOUNT  
GETMODEMNAME  
GETNEXTCOUNTRY  
GETPHONEENTRY  
GETPHONEENTRYCOUNT  
GOSUB  
GOTO  
HANGUP  
HEX  
HOSTECHO  
IF Statement  
IMP  
INCDATETIME  
INKEY  
INPUT  
INSTR

INT  
INTEGER  
KILL  
LASTCONNECTPASSWORD  
LASTCONNEDTUSERID  
LCASE  
LEFT  
LEN  
LET  
LOADPHN  
LOADTRANSLATETABLE  
LOC  
LOCATE  
LOF  
LOG  
LOGFILE  
LONG  
LOOP  
LTRIM  
MAXIMIZE  
MID  
MINIMIZE  
MKDIR  
MOD  
MOUSECLICK  
MOVE  
MSGBOX  
NAME  
NEXT  
NOT  
OCT  
OPEN  
OPENSERIALPORT  
OPENTCPIPPORT  
OR\_Operator  
PAUSE  
PLAY  
POS  
PRINT  
PRINTER  
PUT  
REAL  
RECEIVE  
RECEIVEFILE  
REM  
REMOVEPHONEENTRY  
RESET  
RESETEMULATION  
RETURN  
RIGHT  
RMDIR  
RND  
RTRIM  
SCREEN  
SCROLLBACK  
SCROLLBACKRECORD

SEEK  
SELECT CASE  
SEND  
SENDFILE  
SETCOMM  
SETDTR  
SGN  
SHELL  
SHORT  
SIN  
SIZE  
SOUND  
SPACE  
SPC  
SPLITSCREEN  
SQR  
STAMP  
STATIC  
STOP  
STR  
STRING  
STRING (TYPE)  
STRIPHIBIT  
SUB  
SYSTEM  
TAB  
TAN  
TIMEOUT  
TIME  
TIMER  
TRAPSCREEN  
TYPE\_END\_TYPE  
UCASE  
UPDATEPHONEENTRY  
UPLOAD  
VAL  
VERSION  
VIEWFILE  
VIEWPICTURE  
WAITFOR  
WAITFOREVENT  
WEND  
WHEN\_CLEAR  
WHEN\_DISABLE  
WHEN\_ENABLE  
WHEN  
WHILE\_WEND  
WINVERSION  
XON/XOFF  
XOR

## **+ (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
```

## **\$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
```

## 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,

$\text{abs}(x) = x$       if  $x$  is positive

$\text{abs}(x) = -x$      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 95* the active application on the Windows 95 desktop.

### **Syntax**

ACTIVATE

### **Remarks**

This command will also restore the *QmodemPro for Windows 95* window if it is currently minimized.

### **See also**

[MINIMIZE](#), [MAXIMIZE](#), [MOVE](#), [SIZE](#)

### **Example**

This example minimizes the *QmodemPro for Windows 95* window, waits for the string "connect" from the modem, then reactivates the application.

```
minimize  
waitfor "connect"  
activate
```

## **ADDLFTOCR Function**

Used to get or set the current state of the "AddLFtoCR" 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
```

## **ADDPHONEENTRY Statement**

The ADDPHONEENTRY statement is used to add a new entry to the phonebook.

### **Syntax**

```
ADDPHONEENTRY phoneentry
```

### **Remarks**

The name of every entry in the phonebook must be unique. If the entry name already exists, this statement will fail. The *phoneentry* variable is a variable of type PHONEENTRY. Refer to the appendix for the phoneentry declaration.

### **See also**

[DIAL](#)

### **Example**

This example shows how you might set up the MSI HQ BBS as a phonebook entry and then reset the downloads and uploads to zero.

```
dim entry as phoneentry
entry.NAME      = "MSI HQ BBS"
entry.AREACODE  = "805"
entry.TAPIDEVICE = "Courier HST Dual Standard Fax+ASL"
entry.NIMBER (1) = "873-2400"
entry.USERID    = "Canfield Customer"
entry.PASSWORD  = "QMPROWIN95"
entry.UPLOADS   = 0
entry.DOWNLOADS = 0
entry.CONNECTTYPE = 0 //Set data as connect type
addphoneentry (entry)
```

## 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:

<b>x</b>	<b>y</b>	<b>x AND y</b>
0	0	0
0	1	0
1	0	0
1	1	1

### See also

[EQV](#), [IF Statement](#), [IMP](#), [NOT](#), [OR Operator](#), [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 turn the AutoAnswer mode on or off for a specific modem.

### **Syntax**

```
AUTOANSWER onoff
```

### **Remarks**

If a modem name is passed to this function, autoanswer modem is turned **ON** for that modem. The return value is TRUE if the operation is successful. If AUTOANSWER OFF is used, auto answer mode is turned **OFF**.

### **See Also**

[DIAL](#), [HANGUP](#)

### **Example**

This example tells you whether a key was pressed or a call was answered.

```
if autoanswer (getmodemname (0) ) then
    print "Autoanswer is now on."
else
    print "Failed to configure modem for AUTOANSWER."
```

### **BEEP Statement**

Causes *QmodemPro* to sound the regular Windows 95 beep sound.

### **Syntax**

BEEP

### **Remarks**

If you have attached a .wav file to the "Default Beep" sound in Windows 95 Control Panel, then this command will play the corresponding .wav file.

### **See Also**

[SOUND](#)

### **Example**

This example simply sounds the Windows 95 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. The break signal is often used with older communications hardware, such as mainframes. It usually does not have any effect when connected to a bulletin board or online service.

### **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**

[DIM](#), [INTEGER](#), [LONG](#), [REAL](#), [TYPE](#)

## **Example**

This example declares a variable of type BYTE and assigns a value to it.

```
dim i as byte
```

```
i = 5
```

```
print i
```

## BYVAL Keyword

Indicate the parameter passing method for Subs and Functions.

### Syntax

```
Function Name(BYVAL tStr As String) As String  
Sub Name(BYVAL tNum As Long)
```

### Remarks

The **ByVal** keyword allows you to define how a parameter to be handled within a subroutine at the time of designing the **Sub** or **Function**. The default method for handling parameters is to pass by reference, which means that the subroutine or function can modify the value of the original variable.

By using **ByVal**, it will force the compiler to create a working copy of the data type and leave the passed parameter untouched on return.

### See Also

[FUNCTION](#), [SUB](#)

### Example

```
Rem The difference in passing by value or by reference
```

```
Dim tStr As String
```

```
Function Extend(aStr As String, aLen As Word) As String
```

```
    If Len(aStr) < aLen Then
```

```
        aStr = aStr + String(aLen - Len(aStr), ".")
```

```
    End If
```

```
    Extend = aStr
```

```
End Function
```

```
Function ExtendBV(ByVal aStr As String, aLen As Word) As String
```

```
    If Len(aStr) < aLen Then
```

```
        aStr = Pad(aStr, aLen)
```

```
    End If
```

```
    ExtendBV = aStr
```

```
End Function
```

```
tStr = "Testing this here String!"
```

```
Print "["; ExtendBV(tStr, 60); "]"
```

```
Print "["; tStr; "]" : Rem ByVal = no change
```

```
Print "["; Extend(tStr, 60); "]"
```

```
Print "["; tStr; "]" : Rem By reference = change
```

## 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 optional 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. Since the file name is a string, it will normally be enclosed in quotation marks.

**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 online and connected to another modem.

### **Syntax**

CARRIER

### **Remarks**

The CARRIER function returns TRUE if the modem currently reports that it is online 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 .QSC extension. Scripts are not automatically compiled by this command.

### **See Also**

[END](#), [STOP](#)

### **Example**

This example shows two files, a.QSC and b.QSC, and demonstrates how you can start execution of the second script from the first.

```
[in file a.QSC]
print "in a.QSC"
chain "b.QSC"
```

```
[in file b.qsc]
print "in b.qsc"
```

## 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:\qmw\win" 'change to C:\qmw\win
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 [[#]filename[, [#]filename]...]
```

### **Remarks**

*filename* 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 95* 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
```

### **CLOSEPORT Statement**

This statement closes the currently opened port, if any.

### **Syntax**

```
CLOSEPORT
```

### **Remarks**

CLOSE without any parameters will close all open ports.

### **See Also**

[OPENSERIALPORT](#), [OPENTCPIP](#)

### **Example**

Here is an example of how you might open COM1, send a dial command, and then close the port.

```
if openserialport ("COM1") then print "COM 1 opened."  
SEND "ATDT 1-805-873-2400 ^M"  
Closeport  
print "port closed."
```

### **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 listed in the appendix of this manual.

### **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 scrollbar filename. This is usually the fully qualified path and filename of the scrollbar 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
```

## **COPYFILE Function**

Copy a file to another directory or drive.

### **Syntax**

`COPYFILE (SourceFile, TargetFile)`

### **Remarks**

The **CopyFile** function does just as the name describes, it copies a file from one location to another. It will issue no warning, prior to over-writing a file with an identical name, if one exists in the target directory. The function will return either **True** or **False** upon the success of the operation. Both *SourceFile* and *TargetFile* must include the file name as a path-only in *TargetName* will be rejected.

### **See Also**

[NAME](#)

### **Example**

**Rem** Copy a file across drives

```
If CopyFile("C:WCLIST.OUT", "D:\BAK\DATA\WCLIST.BAK") Then
    Print "File copied"
Else
    Print "Copy failed"
    Beep
    WaitEnter
End If
```

## **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 system format specified in Windows 95 Control Panel, Regional Settings section, short date format.

### **See Also**

[TIME](#)

### **Example**

This example prints today's date.

```
print "Today is "; date
```

## **DATETIMEDIFF Function**

Computes the difference between two variables of type DateTime and returns the result in Days and Seconds.

### **Syntax**

```
DATETIMEDIFF (dt1 as DateTime, dt2 as DateTime, days as integer, seconds as integer)
```

### **Remarks**

The difference is computed without regard to which datetime (dt1 or dt2) is greater. The result will always be a positive number.

### **See Also**

[DATETIME](#)

### **Example**

This example prints to the screen the number of days and seconds between dialing a BBS and hanging up.

```
dim starttime as datetime
dim endtime as datetime
dim totaldays as integer
dim totalseconds as integer

getcurrentdatetime (starttime)
send "ATDT 1-805-873-2400^M"
waitfor "NO CARRIER"
getcurrentdatetime (endtime)
datetimediff (starttime, endtime, totaldays, totalseconds)
print "You were logged on for ";totaldays;"days \
and ";totalseconds;" seconds."
```



## 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. 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 95*, 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"
```

## 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

[ADDPHONEENTRY](#), [DIALNEXT](#), [HANGUP](#), [LASTCONNECTUSERID](#), [LASTCONNECTPASSWORD](#),

### Examples

This example shows each of the four 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**

[ADDPHONEENTRY](#), [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 group box style.

### 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 95 dialog that is created based on this template.

Dialog boxes are discussed in depth in the section titled Using Dialog Boxes.

## **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
```

## 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

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 four 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, IBM3270, RIPSCRIP, TTY, TVI910, TVI912, TVI920, TVI922, TVI925, TVI950, TVI955, VIDTEX, VT100, VT102, VT220, VT320, VT52, WYSE30, WYSE50, WYSE60, WYSE75, WYSE85, WYSE100, WYSE185

### **See Also**

[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:

<b><u>a</u></b>	<b><u>b</u></b>	<b><u>a EQV b</u></b>
0	0	1
0	1	0
1	0	0
1	1	1

### See Also

[AND](#), [IMP](#), [NOT](#), [OR Operator](#), [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](#)

### **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, sr)
```

### **Remarks**

*name* is the filename specification to search for, and can include DOS wildcard characters \* and ?.

*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
```

```
    FileAttributes as long
```

```
    CreationTime as DateTime
```

```
    LastAccessTime as DateTime
```

```
    LastWriteTime as DateTime
```

```
    FileSize as long
```

```
    FileSizeHigh as long
```

```
    FileName as string*260
```

```
    AlternateFileName as string*14
```

```
end type
```

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:\qmwinn" directory.

```
dim sr as searchrec, r as integer
r = findfirst("c:\qmwinn\*.\"", sr)
do while r = 0
    print sr.filename
    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
```

## FORMATDATE Function

Formats a DateTime value into a readable format determined by the picture parameter.

### Syntax

FORMATDATE (picture as string, dt as datetime)

### Remarks

The picture parameter is of the form "MM-dd-yy". Valid items in the picture string are:

<b>Picture</b>	<b>Meaning</b>
d	day of month as digits without a leading zero
dd	day of month as digits with a leading zero
ddd	day of week as a three letter abbreviation
dddd	day of week as a full name
M	month as digits without leading zero
MM	month as digits with a leading zero
MMM	month as a three letter abbreviation
MMMM	month as a full name
y	year as only last digit (if less than 10)
yy	year as last two digits
yyyy	year as full four digits
gg	period/era string

### See Also

[FORMATTIME, TIME](#)

### Example

This example shows how you might use the FormatDate to print the current date as 06/22/95 in the terminal window.

```
dim currentdate as datetime
GetCurrentDateTime(datetime)
print "The date is ";formatdate ("dd/MM/yy", currentdate)
```

## FORMATTIME Function

Formats a DateTime value into a readable format determined by the picture parameter.

### Syntax

```
FORMATTIME (picture as string, dt as DateTime)
```

### Remarks

Valid items in the picture string are:

<b>Picture</b>	<b>Meaning</b>
hh	hour as digits with a leading zero
mm	minutes as digits with a leading zero
ss	seconds as digits with a leading zero
t	time marker string as a single character
tt	time marker as an entire string

### See Also

[FORMATDATE](#)

### Example

This example shows you how to print the current time in the terminal window.

```
dim currentdate as datetime
getcurrentdatetime (currentdate)
Print "The time is ";formattime ("hh:mm:ss tt",currentdate)
```

## **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

[BYVAL](#), [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 [#]filename, [position], variable
```

### Remarks

*filename* 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
```

## **GETCURRENTDATETIME Function**

Gets the current date and time as a DateTime value and places it in the dt variable.

### **Syntax**

```
GETCURRENTDATETIME (dt as DateTime)
```

### **See Also**

[FORMATDATETIME](#), [DATETIME](#)

### **Example**

This example shows you how you might use the GetCurrentDateTime function to print the current time in the terminal window.

```
dim currentdate as datetime  
getcurrentdatetime (currentdate)  
Print "The time is ";formattime ("hh:mm:ss tt",currentdate)
```

## **GETFIRSTCOUNTRY Function**

Fills in a CountryInfo structure with the first country.

### **Syntax**

```
GETFIRSTCOUNTRY (info as CountryInfo) as boolean
```

### **See Also**

[GETNEXTCOUNTRY](#)

### **Example**

This example prints a list of country names seen by Windows 95.

```
type countryinfo
    CountryID as integer
    CountryCode as integer
    Name as string*64
end type

dim cinfo as CountryInfo

if GetFirstCountry(cinfo) then
    do
        print cinfo.name
        loop while getnextcountry(cinfo)
    end if
end if
```

## **GETMODEMCOUNT Function**

Returns the number of modems available in a system.

### **Syntax**

```
GETMODEMCOUNT as integer
```

### **See Also**

[GETMODEMNAME](#)

### **Example**

This example prints the number of modems defined in a system.

```
print "The number of modems configured in this system is "\
    ;getmodemcount
```

## **GETMODEMNAME Function**

Returns the name of modem specified.

### **Syntax**

```
GETMODEMNAME (index as integer) as string
```

### **Remarks**

This function returns the name of the modem specified by the index parameter. The modems are numbered, starting at zero. If GetModemCount indicates there are three installed modems, they are then numbered 0, 1, and 2. The GetModemName function reads these numbers and returns the modem name.

### **See Also**

[GETMODEMCOUNT](#)

### **Example**

Here is an example of how you might use this function to print the names of the modems in a defined system.

```
dim count as integer

for count = 0 to getmodemcount -1
    print "one of the modems is a ";getmodemname (count)
next count
```

### **GETNEXTCOUNTRY Function**

Fills in a CountryInfo structure with information on the next country, pass CountryInfo from a previous call to GetFirstCountry.

#### **Syntax**

```
GETNEXTCOUNTRY (info as CountryInfo) as boolean
```

#### **See Also**

[GETFIRSTCOUNTRY](#)

#### **Example**

This example prints a list of country names seen by Windows 95.

```
type countryinfo
    CountryID as integer
    CountryCode as integer
    Name as string*64
end type

dim cinfo as CountryInfo

if GetFirstCountry(cinfo) then
    do
        print cinfo.name
        loop while getnextcountry(cinfo)
    end if
end if
```

## **GETPHONEENTRY Function**

Retrieves an entry from a phonebook.

### **Syntax**

```
GETPHONEENTRY (index as integer, entry as PhoneEntry) as boolean
```

### **Remarks**

This function retrieves an entry from the phonebook. The phonebook entries are numbered, beginning with zero. If GetPhoneEntryCount returns a value of 10, the entries are numbered 0 through 9. This function returns False if the index is not valid.

### **See Also**

[GETPHONEENTRYCOUNT](#)

### **Example**

This is an example of how you might print the systemnames of the entries in the current phonebook

```
dim count as integer
dim entry as phoneentry

for count = 0 to getphoneentrycount - 1
    getphoneentry (count,entry)
    print ";entry.name" is "System #";count+1;
next
```



### **GETPHONEENTRYCOUNT Function**

Returns the number of entries in a phonebook.

#### **Syntax**

GETPHONEENTRYCOUNT as integer

#### **See Also**

[GETPHONEENTRY](#), [GETMODEMCOUNT](#)

#### **Example**

This example prints the number of phone entries in the current phonebook.

```
print "The number of phonebook entries is ";getphonebookentrycount
```

## **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. The use of the GOSUB and RETURN statements is generally not recommended.

### **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 command tells the system to hang up. The hangup command is sent through Windows 95.

### **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.QSC.

```
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 5 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:

<b><u>a</u></b>	<b><u>b</u></b>	<b><u>a IMP b</u></b>
0	0	1
0	1	1
1	0	0
1	1	1

### See Also

[AND](#), [EQV](#), [NOT](#), [OR Operator](#), [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
```



## **INCDATETIME Function**

Increments a DateTime value.

### **Syntax**

```
INCDATETIME(dt1 as DateTime, dt2 as DateTime, days as integer, seconds as integer)
```

### **Remarks**

Increments the DateTime value dt1 by the number of seconds and days specified the last two parameters, and places the results in the dt2 variable.

### **Example**

This example prints the date seven days from today.

```
dim currentdate as datetime
dim futuredate as datetime

getcurrentdatetime (currentdate)
incdatetime (currentdate,futuredate,7,0)
print "In 7 days, the date will be";formatdate \
    ("ddd, MMMM dd, yyyy",futuredate)
```

## 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 95" within the string "QmodemPro for Windows 95".

```
dim a as string
a = "QmodemPro for Windows 95"
print instr(a, "Windows 95")
```

## **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 -2147483648 to 214783647.

### **See Also**

[BYTE](#), [DIM](#), [LONG](#), [REAL](#), [SHORT](#), [TYPE](#)

### **Example**

This example declares a variable of type INTEGER and assigns a value to it.

```
dim i as integer
```

```
i = 500000
```

```
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**

[DEL](#), [RMDIR](#)

### **Example**

This example removes the file "test.dat" from the current directory.

```
kill "test.dat"
```

### **LASTCONNECTPASSWORD Function**

This function makes the last password entry available.

#### **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 95"  
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 95".

```
dim a as string
a = "QmodemPro for Windows 95"
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 95". See the OPEN script command for an example of how to use LEN with OPEN.

```
dim a as string
a = "QmodemPro for Windows 95"
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 0
```

## **LOADTRANSLATETABLE Function**

Loads a translation table.

### **Syntax**

```
LOADTRANSLATETABLE filename
```

### **Remarks**

This command loads a translation table just like the the Emulation property sheet would.

### **Example**

This example shows how print out verification of a loaded translation table "NewTrans".

```
If LoadTranslateTable ("NewTrans") then  
    print "New Translation Table Loaded."  
else  
    print "Translation Table failed to load."  
End IF
```

## 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 SHORT 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 95  "
```

```
print ">"; ltrim(a); "<"
```

**MAXIMIZE Statement**

Causes *QmodemPro for Windows 95* to maximize its application window on the Windows 95 desktop.

**Syntax**

MAXIMIZE

**Remarks**

The *QmodemPro for Windows 95* application window is maximized and brought to the front of the Windows 95 desktop.

**See Also**

ACTIVATE, MINIMIZE, MOVE, SIZE

**Example**

This example causes the *QmodemPro for Windows 95* 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* (optional) is the number of characters to return from within the string expression. If the number is omitted, the remainder of the string is returned.

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 95"
print mid(a, 7, 11)
```

### **MINIMIZE Statement**

Causes *QmodemPro for Windows 95* to minimize its application window on the Windows 95 desktop. Minimizing an application causes it to appear only on the taskbar.

### **Syntax**

MINIMIZE

### **Remarks**

The *QmodemPro for Windows 95* application window is minimized. Any other Windows 95 opened by *QmodemPro for Windows 95* (phonebook, GIF viewer, editor, and so on) will disappear until the application is restored.

### **See Also**

ACTIVATE, MAXIMIZE, MOVE, SIZE

### **Example**

This example causes the *QmodemPro for Windows 95* application to minimize itself.

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 95* application window on the Windows 95 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 95* 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 Statement](#), [IMP](#), [OR Operator](#), [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
```

## **OPENSERIALPORT Function**

Opens a serial port.

### **Syntax**

OPENSERIALPORT(portname as string)

### **Remarks**

This function opens a serial port. The valid names are generally either "COM1" or "COM2", but you may use other names, depending on the hardware and drivers you have installed. When using this function, do not place a colon (:) after the port name.

If the port is successfully opened, this function returns True.

### **See Also**

[OPENTCPIPSPORT](#), [CLOSEPORT](#)

### **Example**

Here is an example of how you might open COM1, send a dial command, and then close the port. This example includes commands to print a message to the screen when the port is opened and closed.

```
if openserialport ("COM1") then print "COM 1 opened."  
SEND "ATDT 1-805-873-2400 ^M"  
Closeport  
print "port closed."
```

### **OPENTCPIPPORT Function**

Opens a connection using TCP/IP protocol.

#### **Syntax**

OPENTCPIPPORT(host as string, port as integer, telnet as boolean)

#### **Remarks**

This function opens a TCP connection to a specific port on another host using the TCP/IP protocol. The host parameter is either the name of the host (rs.internic.net, for example} or its IP address (such as 198.41.0.6). The port parameter is optional, and defaults to 23, the Telnet port.

The Telnet parameter is also optional, and defaults to True, which means that Telnet protocol will be used during the connection. Passing False as this parameter will open the port in "raw" mode, where no Telnet-specific processing will take place.

If the port is successfully opened, this function returns True.

#### **See Also**

[OpenSerialPort](#), [ClosePort](#)

#### **Example**

Here is an example of how this function could be used to open a TCP/IP port and connect to MSI HQ BBS.

```
if OpenTCPIPport("bbs.mustang.com") then print "Connected to Mustang Software BBS"
```

## 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:

<b><u>a</u></b>	<b><u>b</u></b>	<b><u>a OR b</u></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 95 multimedia services.

### **Syntax**

```
PLAY filename
```

### **Remarks**

If Windows 95 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 95 will not wait for the sound to finish, and will continue execution with the next script command immediately.

### **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 [#]filename, [position], variable
```

### Remarks

*filename* 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

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 four 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
```

```
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."
```

## **REMOVEPHONEENTRY Statement**

Removes a phonebook entry.

### **Syntax**

```
RemovePhoneEntry(entry as PhoneEntry)
```

### **Remarks**

This function removes the phonebook entry specified by the entry parameter. The function returns True if the entry was removed successfully

### **See Also**

[ADDPHONEENTRY](#), [UPDATEPHONEENTRY](#)

### **Example**

This is an example of how you might remove all occurrences of "MUSTANG" from your currently open phonebook.

```
dim count as integer
dim entry as phoneentry
count = 0
do while count < getphoneentrycount
    print "Number of phone entries is ";getphoneentrycount
    getphoneentry (count,entry)
    print "Removing Entry #";count;" : ";entry.name
    if instr (ucase(entry.name), "MUSTANG") then
        if removephoneentry (entry) then
            print "Removed Successfully"
        else
            print "Failed Removal"
        end if
    else
        print "Didn't find Mustang in ";entry.name
        count = count + 1
    end if
loop
```

## 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](#)

### **Example**

This example shows how the RESETEMULATION command might be used.

```
hangup
```

```
resetemulation
```

**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 95"
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:\qmw\test", assuming it is empty.

```
rmdir "c:\qmw\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 95  "
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 width 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 scrollbar 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 scrollbar record mode is turned on. A zero value means scrollbar record mode is turned off.

When scrollbar record mode is on, incoming data is captured to the scrollbar buffer for later review. When scrollbar record mode is turned off, data is not copied to the scrollbar 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 [#]filename, position
```

### Remarks

*filename* 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 Statement](#)

### 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

The first five protocols require a single file name in the filename parameter. This file name indicates the file to send.

The last four 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:\qmw\win\test.dat", Zmodem) = 0 then
    print "file transfer ok!"
end if
```



## **SETCOMM Statement**

Sets the communications parameters for serial ports only without changing the active modem.

### **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.

### **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**

Most modems will disconnect when the DTR signal is lowered. This may be more reliable than using the "hangup" command.

### **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 95 command.

### **See Also**

[END](#), [SYSTEM](#)

### **Example**

This example runs Checkdisk.

```
shell "CHKDSK.EXE "
```

## **SHORT Type**

Used to declare a variable that can handle short numbers.

### **Remarks**

Variables of integer type can hold values that range from -32768 to 32767.

### **See Also**

[BYTE](#), [DIM](#), [LONG](#), [REAL](#), [SHORT](#), [TYPE](#)

### **Example**

This example declares a variable of type SHORT and assigns a value to it.

```
dim i as short
```

```
i = 5000
```

```
print i
```

## **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 95* application window on the Windows 95 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 95* 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

[BYVAL](#), [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 95* application.

### **See Also**

[END](#), [SHELL](#), [STOP](#)

### **Example**

This example terminates *QmodemPro for Windows 95* 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 convert radians to degrees by multiplying by 180/pi (pi is approximately 3.14159).

### **See Also**

[ATN](#), [COS](#), [SIN](#)

### **Example**

This example prints the tangent of 1 radian.

```
print tan(1)
```

## **K + 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 95 Control Panel, Regional Settings section.

### **See Also**

DATE

### **Example**

This example prints the current time according to Windows 95.

```
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 95.

```
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 95"
print ucase(a)
```

## **UPDATEPHONEENTRY Function**

Updates a phonebook entry.

### **Syntax**

```
UPDATEPHONEENTRY (entry as PhoneEntry)
```

### **Remarks**

This function updates the phonebook entry specified by the entry parameter. The function returns True if the entry was updated successfully

### **See Also**

[ADDPHONEENTRY](#), [REMOVEPHONEENTRY](#)

### **Example**

This example changes all 805 area codes to 800 in the currently open phonebook.

```
dim count as integer
dim entry as phoneentry
count = 0
do while count < getphoneentrycount
    getphoneentry (count,entry)
    print "Changing Entry #";count; : ";entry.name;" : ";entry.areacode
    if entry.areacode = "805" then
        entry.areacode = "800"
        if updatephoneentry (entry) then
            print "Update Successful"
        else
            print "Update Failed"
        end if
    else
        print "Didn't find 805 in ";entry.name
    end if
    count = count + 1
loop
```

## 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

The first five protocols require a single file name in the *filename* parameter. This file name indicates the file to send.

The last four 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 95* version number as a string.

### **Syntax**

VERSION

### **Remarks**

In version 2.0 of *QmodemPro for Windows 95*, this function returns the string "2.0". You can always assume that the version numbers will be increasing with respect to the string comparison operators.

### **Example**

This example prints the current *QmodemPro for Windows 95* version number.

```
print "QmodemPro for Windows 95 "; 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](#), [VIEWPICTURE](#)

### **Example**

This example shows how the VIEWFILE command might be used to view the host mode user file.

```
viewfile "host.usr"
```

### **VIEWPICTURE Statement**

Used to invoke the internal picture viewer with a specified file.

### **Syntax**

```
VIEWPICTURE (filename)
```

### **Remarks**

This function brings up the internal picture viewer with the named GIF, BMP, or JPEG 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 to use the VIEWPICTURE function to view a GIF picture in the download directory.

```
VIEWPICTURE ("C:\qmwin\DOWNLOADS\apicture.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"
```

### **WAITFOREVENT Function**

Suspends a script execution until either an incoming call is answered or a key is pressed on the keyboard. It returns an integer, 1 if a key has been pressed, 2 if a call is connected.

#### **Syntax**

```
WAITFOREVENT
```

#### **Remarks**

This function is intended for use by the host mode when waiting for calls.

#### **Example**

This example tells you whether a key was pressed or a call was answered.

```
if autoanswer (getmodemname (0) ) then
    print "Autoanswer is now on."
else
    print "Failed to configure modem for autoanswer."
```

**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 95 version as a string.

### **Syntax**

WINVERSION

### **Remarks**

This function returns the Windows 95 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 95 version 4.0 would be reported as "4.0".

### **See Also**

[VERSION](#)

### **Example**

This example prints out the current Windows 95 version number.

```
print "Currently using Windows 95 "; 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.

### **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:

<b><u>a</u></b>	<b><u>b</u></b>	<b><u>a XOR b</u></b>
0	0	0
0	1	1
1	0	1
1	1	0

### See Also

[AND](#), [EQV](#), [IMP](#), [NOT](#), [OR Operator](#)

### 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
```

## **Contents**

The following Help Topics are available:

[The Picture Viewer](#)

[AutoView during Download](#)

[View Picture](#)

[The File Menu](#)

[The Edit Menu](#)

[The View Menu](#)

[Copying Files](#)

[Printing Files](#)

[Examining Pictures in Detail](#)

[Viewing Multiple Pictures](#)

## The Picture Viewer

You can use the Picture Viewer to look at graphic files. Select the **Tools** menu option, and then **Picture Viewer** to pop up the viewer window. From there, you can select one or more GIF, BMP, or JPEG files to view, either one at a time, or multiple files.

You can also open the Picture Viewer from a command line. Select **Start/Run** from the Windows Taskbar. Type `C:\QMWIN\QMVIEWER`

If you have installed *QmodemPro for Windows 95* to a different drive or directory, type in the appropriate path instead of "C:".



## **AutoView**

The Picture Viewer can show several graphics files in succession. If you are downloading several graphics files, the Picture Viewer will automatically begin to preview the next image when the first file downloaded is complete. You can also choose to view multiple files from the Picture Viewer [File/Open](#) menu command. You can set the time between these viewings by choosing the [View/Options](#) menu in the Picture Viewer. Type in the number of seconds that you wish the viewer to delay between displaying files into the **Delay** box.

## The File Menu

The *QmodemPro for Windows 95* Picture Viewer file menu has options allowing you to open, save, and print files. Here is a list of options, and a brief description of their function in the File menu.

**Open** Opens the Windows common dialog box, allowing you to choose the file you want to open from available drives and directories. The CTRL+O keys will also open this dialog box.

**Open New** Opens a new picture file.

**Print Preview** Shows a representation of the entire picture as it will look when printed.

**Print Setup** Opens the Windows Print Setup dialog box, allowing you to change the setup of your printer, including which printer to use, the printer properties, paper orientation, and paper source.

**Send** Allows you to send the picture file directly to an E-mail message. Opens Microsoft Exchange or your specified electronic messaging system.

## **The Edit Menu**

The *QmodemPro for Windows 95* Picture Viewer edit menu has a copy feature that allows you to mark and copy portions of the currently active picture file to the Windows clipboard.

**Copy** Allows you to copy the selected portion of the picture to the Windows clipboard. Note that when you copy data to the clipboard, it replaces any data currently stored there. You can also use the keyboard command CTRL+C to copy.

**Select All** Selects all the loaded picture files.

## The View Menu

The *QmodemPro for Windows 95* Picture Viewer file menu has options allowing you to open, save, and print files. Here is a list of options, and a brief description of their function in the File menu.

**Toolbar** Toggles the Toolbar Visible command on and off. When **on**, the toolbar is visible from the Picture Viewer window.

**Statusbar** Toggles the Statusbar Visible command on and off. When **on**, the statusbar is visible from the Picture Viewer window.

**Zoom** Expands a portion of the currently active picture for more detailed viewing. You can also use the Zoom icon for this feature. To turn Zoom off, toggle the feature from the View/Zoom menu command or click on the Zoom icon.

**Full Screen** Expands the viewing window to include the entire monitor screen.

**Next** When multiple pictures have been selected, this feature shows the next picture scheduled for viewing.

**Previous** When multiple pictures have been selected, this feature allows you to re-view the picture previously viewed.

**Stop** Stops the slide show

**Options** The Slide Show feature allows you to set the time delay between pictures for viewing. It also gives you the option to repeat the slide show when the last requested picture is viewed.

## **View Picture**

If you have turned on the **View Picture** option in your **Tools/Options/Protocols** setup (from the QmodemPro for Windows 95 main menu), QmodemPro for Windows 95 will automatically open the Picture Viewer when you begin downloading a .GIF, .BMP or .JPG file. If the View Pictures option is turned on, a check will appear beside it. You can view the file as it is downloading. The Picture Viewer allows multiple pictures to be viewed consecutively as they are being downloaded. Use the **Tools/Options** menu choice and select the **Protocols** Sheet to check or change your settings.

## **Copying Pictures**

Part or all of an image can be copied to the Windows 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 toolbar or edit menu (or use the keyboard shortcut, CTRL+C) to put the selected image on the Windows clipboard. Mark the entire image and select **Copy** to grab the image.

## Printing Pictures

If you have a printer connected, you can print the image directly from the viewer by either selecting File/Print, typing CTRL+P, or by pressing the [Print](#) button. The **File** dialog box also offers [Print Preview](#).

## **Examining Pictures in Detail**

If you want to examine a part of the picture in more detail, you can zoom in by clicking the **Zoom** button on the toolbar. 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. Clicking on the Zoom button again returns the file to its original image.



## Opening More Picture Files

To open another file or group of files from the Picture Viewer, select the **File/Open** menu, or you can click on the file you want to view in Windows 95 Explorer. You can also drag one or more file icons from the Explorer into the Viewer window.

## **Contents**

The following Help Topics are available:

[About MSI](#)  
[Top Ten Cool Things About QmodemPro for Windows 95](#)  
[Ordering Information](#)  
[Order Form](#)  
[Last Minute Changes](#)  
[New in QmodemPro for Windows 95](#)  
[QmodemPro for Windows 95 Tutorial](#)  
[Using Command Lines to Run QmodemPro for Windows 95](#)  
[The Phonebook](#)  
[The Editor](#)  
[The Picture Viewer](#)  
[The Scripts Menu](#)  
[The Host Mode](#)  
[Macros](#)  
[Programmable Keys](#)  
[Troubleshooting Guide](#)  
[Customizing the Desktop](#)  
[Troubleshooting](#)  
[Customer Access](#)  
[Modems](#)  
[The Main Menu](#)  
[The File Menu](#)  
[The Connection Menu](#)  
[The View Menu](#)  
[The Script Menu](#)  
[The Tool Menu](#)  
[The File Clipboard](#)  
[The Statusbar](#)  
[Customizing the Toolbar](#)  
[Last Minute Changes](#)

## **Host Mode**

The Host is a small-scale BBS program that allows you to set up your computer to receive incoming modem calls. Callers can read and post messages, upload and download files, and if you allow it, take control of your computer to run simple Windows commands.

**For more information on using the Host Mode, click on one of the following topics:**

[Starting Host Mode](#)

[Setting Up](#)

[Running Host Mode](#)

[Common Commands](#)

[Self-Created Files](#)

[Menu Display Files](#)

[Help Files](#)

[Colors](#)

## **Setting Up**

The setup for QmodemPro Host contains only a few options. However, some planning and organization is required. We recommend you read this section completely before beginning the setup, so that you understand the implications of each option.

[Mode](#)

[Security-Related Options](#)

[File Transfer Options](#)

## Running Host Mode

At all times, Host Mode will accept input from the remote caller as well as from the local keyboard. The sysop, for example, may come to the rescue of a caller by typing commands directly. These keystrokes will be processed as if they had been typed by the remote caller.

Press F1 to log on to QmodemPro Host locally. All Host commands except Upload and Download are available in a local logon. When you are finished your local logon session, use the **Goodbye** command to log off. When you log on locally the Busy command is sent to the modem.

You can chat on-line with a caller by pressing the F2 key. This key is also used to answer a page from a caller. Both caller and Sysop may type at will; the display will look the same on both computers. End-of-Line wordwrap takes place automatically in CHAT mode. Press F2 again to exit CHAT mode. The message "Returning you to Host Mode" will be displayed and the caller will be returned to the Main Menu.

Use the **Cancel** button on the file transfer status window if you need to abort a caller's upload or download without waiting for the file transfer to finish.

You may for any number of reasons need to disconnect a caller immediately from the QmodemPro Host. The **Hangup** button or **Connection/Hangup** command on the menu drops the connection instantly, with no notification to the caller.

## Commands

These commands will be available to callers when they are logged on to your system while you are using the QmodemPro Host:

**R** - [Read Messages](#)

**E** - [Enter a Message](#)

**P** - [Page the Sysop](#)

**F** - [File Directory](#)

**D** - [Download Files](#)

**U** - [Upload Files](#)

**G** - [Goodbye](#)

**Z** - [Shut Down Host](#)

**S** - [Shell to a Command Line](#)

## Host Mode Self-Created Files

Two files are created by the Host the first time it is run:

**HOST.HDR**     The Message Header file.

**HOST.MSG**     The Message Text File

These two files are maintained by QmodemPro Host. Do not try to edit or modify these files in any way, or system errors may occur when callers attempt to read or enter messages.

The only maintenance required by the system operator on the message base is periodic deletion of killed messages. Press F7 at the Waiting for Calls screen to pack out these deleted messages. The deleted message space will be removed from both HOST.MSG and HOST.HDR, reducing their size.

## **Menu Display Files**

The following menu display files are included with the Host Mode Script

### **HOST.PRE**

The opening screen sent to all callers.

### **HOST.MNU**

The Host Main Menu.

### **HOST.PRO**

The protocol list for Up/Download commands.

### **HOST.OFF**

The logoff screen sent to all callers when they say Goodbye.



## Help Files

The following help files are included with the Host Mode Script:

### **HOST.HLP**

Generic Help about the Host, including a brief command summary.

### **HOSTD.HLP**

Help on the Download command. You can edit this file to provide additional information, if desired.

### **HOSTE.HLP**

Help on the Enter Message command. Instructions on entering a message.

### **HOSTF.HLP**

Help on the File List command. Tells the caller about your downloadable files.

### **HOSTG.HLP**

Help on the Goodbye command.

### **HOSTP.HLP**

Help on the Page Sysop command. This explains what hours the sysop is available for paging, or instruct the caller to leave a message if the sysop is not available to chat.

### **HOSTR.HLP**

Help on the Read Message command. Instructions on reading messages.

### **HOSTS.HLP**

Help on the Shell to DOS command. Instructions on executing supported DOS commands, and how to return to the host when finished.

### **HOSTU.HLP**

Help on the Upload command.

### **HOSTZ.HLP**

Help on the Shutdown Host command.

## **Colors**

QmodemPro Host supports colors in display files using the standard ANSI color codes. These codes may be placed anywhere desired to alter the color of the display. However, if you use ANSI codes in your display files, all callers must have ANSI emulation active to view the screens correctly. Callers without ANSI capabilities will see the codes rather than the colors.

## Using QmodemPro for Windows 95 with Command Lines

To access a command line, from the main Windows screen, click on **Start** and choose **Run ...**, or click on the DOS macro on the default macrobar at the bottom of the QmodemPro for Windows 95 terminal screen.

To see how to use a command line, click on the specific command you are interested in:

[Open QmodemPro for Windows 95](#)

[Dial into a telnet site](#)

[Dial a group entry](#)

[Dial using a different phonebook](#)

[Dial an entry and automatically execute a script on connection](#)

## **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.

For more information, click on one of the topics below.

[Macros](#)

[Keymaps](#)

## The File Clipboard

The QmodemPro for Windows 95 File Clipboard lets you mark file names on the screen and store them for use later without ever typing them in. To open the File Clipboard, click on the **File Clipboard** icon on the toolbar, or use the **View/File Clipboard** menu command.

During an online session, open the clipboard and mark the filename on the terminal screen and drag it to the clipboard edit box. Qmodem automatically copies the filename to the File Clipboard.

Using the **Look Up** or **Look Down** buttons, you can search long lists of files in the terminal window, and choose the files you want. These two commands search for text followed by "." (a dot).

To grab and edit a filename, double click on the file name to highlight it, or highlight it with the mouse, then drag it to the File Clipboard.

If you want to save the filename exactly as it is, just grab it and drop it into the lower box in the File Clipboard. This is where lists of filenames are stored.

If you need to edit the file name, drag it to the upper box in the File Clipboard. This section is the edit box. From here, you can edit the selection. Once you are satisfied with the selection, click **Add** and the filename will be added to the list in the box below.

The **Save As** button lets you save lists of files as a Qmodem File List. To remove a file from the file list, mark it and click **Remove**.

**Send** sends all marked text to the terminal screen. If you want to download a list of files, you can bring up the list you want, using **Open** to load a list of previously saved filenames such as "Spreadsheets" into the file clipboard. Choose **Send** to send the entire list of filenames to the terminal. You don't have to type each file name separately.

If you check the **Remove Items on Send** feature, the file clipboard will automatically remove the file names once they are sent to the terminal.

The **Send Separator** allows you to tell the File Clipboard which character to send to the terminal between filenames. This can be carriage return, a line feed, a space, a comma, colon, semicolon, or tab. The send separator is important, because when you download files from the host program, you will need to tell the host how to see each filename. Some hosts look for a carriage return, some for a space, etc.

**Long Filenames** lets the File Clipboard to take advantage of Windows 95's ability to use filenames longer than 8 characters.

## Key Maps

Up to 256 reassignments may be programmed for each terminal emulation, consisting of 12 function keys and 32 additional keys in normal, shift, alternate and control positions.

For most users, the default settings provided are correct. We recommend that changes not be made in the Emulation keymaps unless it is absolutely necessary. Changes to the keymap structure will affect the emulation.

To create and edit KeyMaps, go to the **Tools/Options/Emulations** property sheet, 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.

## Starting Host Mode

### To start the Host Mode:

From the terminal screen, select **Scripts/Host Mode Script**. QmodemPro for Windows 95 will automatically look for and execute HOST.QSC. If you start QmodemPro for Windows 95 with the command line parameter HOST.QSC, the program will automatically start up in host mode and be available for incoming calls.

It is possible to start the Host while you are connected by modem to another computer. The other person you are connected to will be prompted for logon information as if the host had answered the incoming call itself.

If the caller is unsuccessful in his logon, the host will drop the connection and cycle. You may return to on-line terminal mode at any time during host operation by stopping the script.

To stop a script, select **Scripts/Execute**, or click the **Execute Script** button. This will toggle the script execution **off**.

## Modes

The first configuration item is **Mode**. There are three selections: Open, Closed, and Callback.

An **Open** system is the simplest to use and maintain. Anyone who calls your Host can log on and register as a new user. As new callers log in, the Host automatically adds them to its user record file (HOST.USR). All new users are assigned a security level of 0 (zero).

You may, however, want to restrict access to only those callers you have authorized. In a **Closed** system, only callers whose names and passwords are already defined in the HOST.USR file can log on. If the caller's name is not on file, the Host displays a brief message that the caller is not registered, and hangs up the phone.

A **Callback** system is similar to a Closed system in that it requires the callers' names and passwords to be in the HOST.USR file. In addition, CALLBACK requires that the caller's DATA Phone Number also be on file. When callers log on and successfully enter their password, they will be prompted to hang up and place their modem into Auto-Answer mode. The Host will then hang up and immediately dial the caller's data phone number. A Callback system is very secure since it insures that all communications originate from known locations.



## Security-Related Options

**Max Time** This defines the maximum time allowed per call, in minutes for all callers. The default value is 60 minutes. When a caller's time expires he is logged off the system, after a warning.

**DOS Password** The password required to execute Windows command line commands from remote. The default here is blank, which disables the command lines option completely. To enable command line commands, enter a password in this field.

If enabled, this option allows your callers to copy, list, even delete files! For obvious reasons, the password you select for this option should be one not easily guessed by callers.

**Shutdown Password** To execute the **Shutdown Host** command from a remote connection, the Host requires a password. As with the DOS Password, this option is disabled by default, and can only be enabled by entering a password in this field. If the correct password is entered, the system hangs up immediately and Host mode is terminated. For obvious reasons, the password you select for this option should be one not easily guessed by callers.

## File Transfer Options

**File transfers** You may wish to make file transfers available to your callers. The following options control which subdirectories callers can use for file uploads and downloads. By default, both these paths are blank, disabling uploading and downloading.

**Download Path** All files available for download from the Host must be stored in this directory. The download path contains files that can be transferred from your host to the caller's system. Any valid directory may be entered in this field. For example, C:\PROGRAM FILES\MUSTANG SOFTWARE\QMODEMPRO\HOST\DOWNLOADS. Only one path name may be entered, and it is the only path for downloads. Files located in subdirectories of the download path are not available to callers.

The Host **File List** command will list all the files in this directory. Do not place files in this directory unless you want to make them available to callers.

**Upload Directory** All uploaded files are stored into this directory. The upload path is the location where all files sent to you by your callers are stored. Any valid DOS directory may be entered in this field. For example, C:\PROGRAM FILES\MUSTANG SOFTWARE\QMODEMPRO\HOST\UPLOADS. If the Upload and Download directories are the same, all uploaded files become public and are available for download.

**Sysop can download from any path** Check this option to allow those with Sysop security to specify a full drive, path and file name. This allows such callers to download any file on the system, if the caller has a security level greater than 0, and if the complete file path and name are entered. Note that only the path specified will be listed when the F List command is used, so to use this sysop over-ride the exact path and file name must be known already. You should exercise discretion in allowing sysop security to other users of your Host.

**Modem** The **Modem** drop down box in Host Setup allows you to select one of the modems detected by Windows 95 for the Host to use to answer incoming calls. QmodemPro for Windows 95 allows you to directly connect to the modem. If you do not have the correct TAPI drivers, you may want to use direct connect.

## Read Messages

The **Read** command is used to read messages. Unlike more advanced On-Line Services or Bulletin Board Systems, the Host does not keep track of each caller's Last Message Read. It is up to callers to remember the number of the last message they read.

If the caller presses [ENTER] at the "Start with Msg" prompt, the READ command is canceled and the Main Menu is displayed.

If the number entered is lower than the number of the first message, reading will start with the first available message. Likewise, if the number entered is higher than the highest message number, the last valid message will be read.

Message headers contain **Message Number, To, From, Subject, Received** and **Date & Time** information. Message marked ""(private)" or "(killed)" can only be viewed by callers with SYSOP Security or by the caller who left the message.

After reading a message, the caller is prompted for one of the following actions:

[N] to read the next message.

[R]Reply to the message just read.

[K]Kill the message (Sysop or recipient only).

[Q]Quit and return to the Main Menu.

## **Page the Sysop**

When someone **Pages the Sysop**, you and the caller will hear a series of beeps, and the following message will appear on the screen:

Paging Sysop. Press any key to abort...

The sysop's screen shows an additional message, instructing the sysop to press F2 to answer the page and begin chatting with the caller. If the page is not answered by the system operator after about 20 seconds, the Host will terminate the page, inform the caller that the sysop is unavailable, and return the caller to the Main menu.

## **Enter a Message**

The **Enter a Message** command is used to create a new message. The caller will be prompted for the name of the recipient, the subject of the message, and the security status of the message (private or public). The caller is then placed in the message editor.

Messages may have up to 99 lines of 72 characters each. As text reaches the end of the line, it automatically wraps to the next line.

At the end of the message, the caller presses [ENTER] twice to terminate message entry. The Host will then display the (C)ontinue, (S)ave or (A)abort prompt.

The last line with text is displayed with the cursor following the last non-blank character. Use this option if you exited the message editor and wish to continue typing.

The message is saved and the user is returned to the Main Menu. If the message was entered as a reply, the next message, if any, will be displayed. The message is deleted and the user is returned to the Main Menu. If the message was entered as a reply, then the next message, if any, will be displayed. The end of message prompt will also be generated when the message reaches the 99-line maximum.

## **Files Listing**

The **Files Listing** command lists the contents of the Host Download Directory. If the directory exceeds one page, the display is paused, and the caller is prompted to continue, stop the display, or show the entire list nonstop.

## **Download Files**

The **Download** command lists the Protocol menu file HOST.PRO if found, otherwise the default list of protocols is displayed.

The caller can then select the desired protocol and is then prompted to enter a filename. During the transfer, the standard QmodemPro for Windows 95 send file status window will be displayed.

Before the download begins, the Host checks for the existence of the file on disk. If the file does not exist, the caller's Download request is canceled.

If a batch protocol is selected, the user is prompted for up to 20 filenames to download. Filenames can contain the wild card characters \* and ?.

## **Upload Files**

When the **Upload** command is selected by a caller, the Host first looks for the alternate protocol menu file PROTS.HST. If it is found, it is displayed. If not, the list of default protocols is displayed.

The caller can then select the desired protocol. A filename prompt will be displayed next for the file transfer.

During the transfer, the standard QmodemPro for Windows 95 receive file status window will be displayed. The meanings of Uploads and Downloads are reversed while in Host Mode. An Upload by the caller is a Download to the Host, and vice-versa.



## **Goodbye**

The **Goodbye** command logs off a caller and drops the modem connection. The modem goes back on-hook and QmodemPro Host returns to the "Waiting for call..." screen.

The "?" command will display HOST.HLP if found. This is a generic **Help** file that should tell the caller what all the commands are and briefly what they do. It should also mention that more help is available by typing in a ? followed by the command in question. This will display a file by the name of HOST[character].HLP where [character] is the command letter they want more help with. Sample Help files are included, and may be modified, if desired.

## **Shut Down Host**

The **Shutdown** command can only be entered by users that have sysop security (security level 1). In addition, they must also know the shutdown password. When the correct password is entered, the Host hangs up and returns to the QmodemPro for Windows 95 Terminal screen. For security reasons this option is not shown on the default menus.

To exit Host Mode, press F9 while the system is waiting for a call or click on the Script icon on the toolbar. You will be returned to the Terminal window.

## **Shell to DOS**

The **DOS Prompt** command allows callers to use internal Windows commands to list directories, copy existing files, rename files, delete files, change directories, make directories, remove directories and type text files to the screen.

The caller will be prompted for a password, and if the password is entered successfully, the Host will allow access to these Windows commands. This prompt is not actually a link to DOS on the Host. It is an internal Host function that looks like DOS and passes the commands to the Host PC system. Programs cannot be run from remote. Activities are limited to the commands specified above.

## **Open QmodemPro for Windows 95**

The simplest command line switch simply starts up QmodemPro for Windows 95, specifies the phonebook file to load, and dials a specific entry. Entries are specified by name. For example, to dial the MSI HQ BBS in the default Phonebook file, the command line looks like this:

```
C:\PROGRAM FILES\MUSTANG SOFTWARE\QMODEMPRO "MSI HQ BBS"
```

### **Dial into a Telnet Site**

To manually dial a telnet site, add the TELNET command to the command line string. For example, if you want to telnet to the MSI HQ BBS, you would type

```
QMWIN TELNET BBS.MUSTANG.COM
```

in the command line. This lets you use QmodemPro for Windows 95 as your default telnet dialer for use with a Web browser.

### **Dial from a Different Phonebook**

If you have a phonebook file named NANCY.PHN containing only Nancy's phone numbers. To start up QmodemPro for Windows 95 with your NANCY.PHN Phonebook loaded, use a command line like this:

```
QMWIN NANCY.PHN
```

### **Execute Scripts Automatically at Opening**

To execute a script automatically when you start QmodemPro for Windows 95, simply specify the name of the script on the command line, like this: QMWIN SCRIPT.QSC **or** QMWIN C:\PROGRAM FILES\MUSTANG SOFTWARE\QMODEMPRO\SCRIPTS\SCRIPT.QSC Unless a full path is included, the script file must exist in the default script directory, specified in the **Options/Files/Paths** property sheet, and the extension .QSC or .QSX is required.

## Dialing a group

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.

If you want to send a command line that will tag and dial every system in the group named "Computer Groups" and the group is in the default phonebook, you can call the group from a command line by typing:

```
QMWIN GROUP "COMPUTER GROUPS"
```

If the group is in a different phonebook, you need to add that information to the command line. For example, if the group named "Computer Groups" is in a phonebook named BRIAN.PHN, you would type:

```
QMWIN BRIAN.PHN GROUP "COMPUTER GROUPS"
```

This works in the same way as the Search menu command in the Phonebook. Load the Phonebook command line along with the text you want to find. You must put quotation marks around the search text. For example, if you want to dial everything in the ONLINE.PHN phonebook matching the text "service", you would type the command line in like this:

```
QMWIN ONLINE.PHN "service"
```



## **Customer Access**

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 through our BBS or by phone.

[Benefits of sending your registration](#)

[Technical Support Services](#)

[MSI HQ! BBS](#)

[CompuServe](#)

[America Online](#)

[GEnie](#)

[Internet](#)

[WWW](#)

[Telnet](#)

[MSI SupportNET](#)

[Voice Support](#)

[Microsoft Network](#)

[Questions other than technical support](#)

**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
- 24-hour access to the private multiline Registered Users 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 Services**

QmodemPro for Windows 95 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 F1 help key for tips and explanations. If you're still stumped, technical support is only a phone call away.

**MSI HQ! BBS**

You may 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.

**CompuServe**

On CompuServe, we are a part of the PC Vendor Support Forum. You reach us by typing GO PCVENA and then selecting Subtopic 9.

**GENie**

You can find us on GENie, the on-line service from General Electric. Go to the MUSTANG Round Table with the command MOVE 680.

**America Online**

MSI's support section on America Online is called MUSTANG and can be reached with the keyword MUSTANG.

**Internet**

Our Internet mailing address for tech support is [support@mustang.com](mailto:support@mustang.com)



**Telnet**

To telnet into the MSI HQ BBS, telnet to [bbs.mustang.com](http://bbs.mustang.com)

**Microsoft Network**

Click on the "MSI on MSN" icon included in your program menu.

## **Voice support**

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 which is bound inside your QmodemPro for Windows 95 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 directly at (805) 873-2550.

## **MSI SupportNET**

We also have a large number of BBS systems carrying the MSI SupportNET echo conferences throughout the US and Canada, and around the world. These BBSs are located in Australia, the United Kingdom, France, and Norway.

The MSI SupportNET contact points outside the US are:

<b>Location</b>	<b>Distributor</b>	<b>BBS Phone</b>
Australia	Banksia Info. Tech.	+61 2-418-7693
France	Maxotex	+33 1-45-34-6430
Scandinavia	PC Security	+47 2-58-3358
United Kingdom	Telesystems	+44 49-489-1903

### **Questions Other Than Technical Support**

If you have questions related to shipping or other areas handled by our office staff you should contact us at 805-873-2500. Our office staff can help with changes to your registration information and handle all non-support questions regarding our product line.

We also have a toll-free number for questions regarding new product purchases or updates you can call 1-800-208-0615, Monday to Friday, 8:00 a.m. to 5:00 p.m. Pacific time. Our mailing address is Mustang Software, Inc., PO Box 2264, Bakersfield, CA 93303.

**WWW**

You can visit our home page at <http://www.mustang.com>

## **How to Reach MSI**

We can be reached in any of the following ways:

Mustang Software, Inc.  
Post Office Box 2264  
Bakersfield, CA 93303-2264

Business Office (805) 873-2500    ORDERS ONLY (800) 208-0615  
Internet: [sales@mustang.com](mailto:sales@mustang.com)  
World Wide Web: <http://www.mustang.com>  
BBS: 805-873-2400

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.

Technical support for *QmodemPro for Windows 95* is available by phone Monday through Friday 9 am to 5 pm Pacific time. To reach the MSI Technical Support staff, call (805) 873-2550.

## **Last Minute Changes**

Software is constantly undergoing refinements and minor changes. Before *QmodemPro for Windows 95* was released, a few of these changes were made. These changes were made too late to appear in the printed documentation. Here is a list of the changes made since the time that 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.

[Last Minute Changes to Qmodem](#)

[Last Minute Changes to the Qmodem Editor](#)

For Help on Help, Press F1



## Changes to Qmodem

Software is constantly undergoing refinements and minor changes. Before *QmodemPro for Windows 95* was released, a few of these changes were made. These changes were made too late to appear in the printed documentation. Here is a list of the changes made since the time that 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 SERVICES.QSC script no longer exists. It has been separated into four separate script files. These files are:  
MAKECIS3.QSC - Script for automatic logon to CIS (CompuServe)  
MAKEDEL3.QSC - Script for automatic logon to Delphi  
MAKEMCI3.QSC - Script for automatic logon to MCI  
MAKEGEN3.QSC - Script for automatic logon to GENie
- Added new Direct Dial option to the phone entries in the phonebook, this will allow you to dial without having TAPI do translations for you. This feature is mostly for dialing into PBX systems, etc. This means that the area code and country code are not mandatory, if the Use Country Code and Area Code selection is turned off. This new feature is accessed through the dialing directory property sheet entry "Use Country Code and Area Code".
- If you change the com settings in the phonebook using the Configure button they will now stick for that entry.
- If you have an entry in the phonebook set to data call and has a script attached but no area code or phone number QmodemPro will just run the script.
- Added new print field of Area Code to display the area code.
- Modified the Direct Connection sheet, the property sheet now searches for all available ports. If you have other COM ports you want available, you can add them to the registry in:  
HKEY\_CURRENT\_USER\\Software\\MSI\\QmodemPro\\Direct Ports
- Ports should be new string items with the name of the COM port (such as COM15).
- Added a Last Call Duration option on the Statistics property sheet of Edit Phonebook Entry. This feature displays the length of time the last connection was active to this phonebook entry.
- The Tools/Options/Sounds textbox now allows you to scroll horizontally, so that you can add sound names longer than the visible text.
- Dial out Telnet by port is now available. To add a port to a telnet address, add ":xx" (where xx is the port number) to the end of the address. For example, to telnet to port 33 at the MSI HQ BBS, you would type the address as:  
bbs.mustang.com:33
- You can now configure telnet connections to force Binary Telnet mode.
- The dialing directory now updates the modification flag when adding groups, and saves the phonebook properly.
- The file transfer status box no longer "wraps" the elapsed time on longer file transfer times.
- You can now add path names longer than 30 characters to the download path dialog box.
- Manual dial now resets the last dialed entry.
- Manual dial now displays the last 5 manually dialed numbers when you click the down arrow beside the entry text box.
- Long file names are now passed properly from QModemPro to the Qmodem Editor.
- When uploading files, the CPS display now reports the correct CPS.
- When an exit sound has been associated with QmodemPro, it is now played correctly before the exit occurs.
- The printer routines now recognize long file names, allowing you to print to PRN when installed to a long file name directory.
- The dialing directory now arranges icons properly.
- Fixed a bug in the RIP routines which caused an ObjectWindows error.

## **Changes to the Editor**

Software is constantly undergoing refinements and minor changes. Before *QmodemPro for Windows 95* was released, a few of these changes were made. These changes were made too late to appear in the printed documentation. Here is a list of the changes made since the time that the manuals were printed. In all cases the on-line help file (QMEDITOR.HLP) has been updated in the appropriate places to reflect these changes.

## **The Statusbar**

The statusbar displays current system information. The first field displays the name of the phonebook entry and connection rate. The second field shows the current terminal emulation. The third field lets you know when you are online or if you are offline. The fourth field shows the name of scripts that are active with this phonebook entry. The last field shows the current time if you are offline, or the amount of time you've been connected if you are online.

## **Top Ten Cool Features in QmodemPro for Windows 95 v2.0**

### **10. RIP Graphics terminal emulation support gives you a Point & Shoot GUI interface on thousands of BBSs around the world.**

RIP stands for Remote Imaging Protocol and it's the graphic terminal emulation standard supported by Wildcat! BBS, Major BBS, PC Board, TBBS and several other popular BBS software platforms. When you dial a RIP-capable BBS you can navigate with your mouse and enjoy the eye-popping graphics. It's the ultimate interface for the BBS surfer!

### **9. Powerful 32-bit, multithreaded design lets you take full advantage of Windows 95's multitasking.**

To benefit from the 32-bit horsepower provided by Windows 95 you need applications specifically designed to harness it. Using your old 16-bit communications software in a 32-bit environment will work, but it will adversely affect the performance of the entire system. QmodemPro delivers advanced features and performance in a specifically designed 32-bit package that keeps other Windows 95 applications humming.

### **8. MAPI links allow you to send session text, images and files to other MAPI-aware applications.**

It's easy to send any marked text, capture files or even the scrollback log to another user via any MAPI (Mail Application Programming Interface) compatible application like Exchange, MS Mail, cc:Mail or Notes.

### **7. Use QmodemPro as a full-featured Telnet client.**

Telnet into computers and BBSs around the world using all the features you're used to in a dial-up connection -- robust terminal emulations, speedy file transfers, and customized keyboard maps and macros. Plus, QmodemPro supports RSA Data Security, Inc. MD5 Message-Digest Algorithm for password security.

### **6. The most powerful and flexible phonebook you've ever seen!**

QmodemPro gives you a true Windows 95 phonebook. Each entry can have its own icon and you have four ways to view the phonebook (Large Icons, Small Icons, List and Details). The phonebook also supports up to 5 numbers per entry, stores a unique user ID and password for each entry, supports data, telnet and voice in the same phonebook and can be configured to load automatically when QmodemPro for Windows is started.

### **5. QuickLearn script recorder makes it super simple to automate repetitive actions such as logging onto a BBS or online service.**

Writing scripts has never been easier! All you need to do to have QmodemPro create a logon script is define a script name in the phonebook and dial the number. QmodemPro will automatically toggle into QuickLearn mode, record your logon and create a script for you to automate future logons.

### **4. The SLIQ compiler and debugger give you all the tools you need to develop custom applications in SLIQ (Script Language Interface for QmodemPro).**

SLIQ is similar to the popular BASIC programming language so it's easy to learn and use, yet powerful enough for the advanced user to develop specialized applications. QmodemPro's Host Mode script provides an example of the power and flexibility of SLIQ.

### **3. TAPI support means QmodemPro will share your modem with other TAPI-compliant applications.**

You can run QmodemPro concurrently with the Microsoft Exchange Fax system, Dial-up networking, Microsoft Network, or any other TAPI (Telephony Application Programming Interface) compatible application without having to worry about port conflicts.

### **2. OLE 2.0 Drag & Drop support lets you drag phonebook entries right to your desktop.**

You can easily create shortcuts with QmodemPro's Drag and Drop support. Commonly dialed numbers are then just a mouse-click away, even if you don't have QmodemPro currently running. You can also drag & drop files directly from the Windows Explorer into QmodemPro's file transfer window. Just mark the files you wish to upload in the Windows Explorer and just drag them to the Send Files window in QmodemPro. You can even mark the files before connecting to save time online.

### **1. GIF/BMP/JPEG Picture File Viewer (PFV) that lets you zoom any graphic file up to 1600%, even while the file is still being downloaded!**

Just click the Zoom button on the toolbar and then left-click to zoom in and right-click to zoom out. You can even print or cut and paste the image to other applications. Way cool!

## Ordering Information for QmodemPro for Windows 95

Check out the [Top Ten Features](#) to find out why QmodemPro for Windows 95 v2.0 is *The Premier Communications Solution for Windows 95*.

QmodemPro is available direct from Mustang Software, Inc. for \$129.00 plus shipping and handling. You can order QmodemPro by calling toll-free +1 (800) 208-0615 or +1 (805) 873-2500, or order by fax or mail by printing out this handy [Order Form](#).

You can also purchase QmodemPro for Windows 95 v2.0 at your local CompUSA, Computer City, Software Etc, Babbage's, Egghead Software, Incredible Universe, Best Buy, Office Max, Office Depot, Electronics Boutique, Elek-Tek, Fry's Electronics and other fine software stores everywhere.

When you purchase QmodemPro you will receive:

- Comprehensive 300+ page User Guide and 325+ page Script Guide
- Registered user access to the MSI HQ BBS
- Free technical support by phone, MSI HQ BBS, Compuserve, America Online and Genie
- Free subscription to the MSI Monitor Newsletter
- Special discounts and offers on selected MSI products
- Plus much, much more!

## QmodemPro for Windows 95 v2.0 Order Form

**Yes, I want The Premier Communications Solution For Windows 95!** Send me QmodemPro for Windows 95 v2.0 for only \$129 plus shipping.

Please print your name and shipping address below (No PO Boxes, please):

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

City: \_\_\_\_\_ State/Province: \_\_\_\_\_

Zip/Postal Code: \_\_\_\_\_ Country: \_\_\_\_\_

Daytime Phone number: \_\_\_\_\_

### Payment method:

\_\_\_\_\_ Check/Money Order (US Funds Only, Please)

\_\_\_\_\_ Credit Card

Visa, MC, Discover \_ \_ \_ \_ \_ " \_ \_ \_ \_ \_ " \_ \_ \_ \_ \_ " \_ \_ \_ \_ \_

American Express \_ \_ \_ \_ \_ " \_ \_ \_ \_ \_ " \_ \_ \_ \_ \_

Signature: \_\_\_\_\_ Expiration Date: \_\_\_\_\_

Note: Be sure to include shipping and applicable taxes. Shipping for US customers is \$10. International customers contact Mustang Software, Inc. (800-208-0615 or 805-873-2500) for shipping options and costs. California residents must include local sales tax.

### To Order:

#### By Phone:

Please have your credit card and order ready.

Inside US and Canada, call Toll Free: (800) 2080615

Outside US: (805) 873-2500

#### To Fax Your Order

Our fax number is (805) 873-2599

If you're paying by credit card, you may also fax this entire form if you wish. Orders faxed on weekends and holidays are usually processed the next business day.

#### To Mail Your Order

Send your check, money order, or credit card information with this form to:

Mustang Software, Inc.  
PO Box 2264  
Bakersfield, CA 93303-2264

{button Print Order Form,Print() }

Q95 Demo 11/3/95

## **A Guided Tour of QModemPro for Windows 95**

Take a few minutes to get some practical experience with QmodemPro for Windows 95. It's not required, it's just a fast, easy way for you to become familiar with QmodemPro for Window 95 and with Bulletin Board Systems.

Mustang Software, Inc. operates a Wildcat! BBS (Bulletin Board System) to support our customers and provide access to files and messages that can help you use QmodemPro for Windows 95. We call it the MSI HQ BBS.

[Dial the MSI HQ BBS](#)

[Connecting](#)

[Logging on](#)

[A Common Password Problem](#)

[New User Information](#)

[Exploring the BBS](#)

[Downloading](#)

[Expanding Your File](#)

[A Note About Compressed Files](#)

[For More Information](#)

## The Terminal Window

This is the main screen of QmodemPro for Windows 95. The top section of the Terminal window is the Command Menu. Beneath that is the Toolbar. The main screen, below the toolbar, is where all your online data will appear. Beneath the main screen is the Macrobar Menu, and below the Macrobar menu is the Statusbar.

QmodemPro for Windows 95's terminal window is, in a sense, the main control panel used to communicate with your the modem and, once you're on-line, to interact with the remote modem.

This window is surrounded by QmodemPro's main screen which provides information about the status of your connection, a menu bar with command options, a toolbar with buttons for frequently-used functions, and at the bottom of the window, a macrobar and a status line, showing current parameters. You can customize your toolbar or hide it. Click on the toolbar with the right mouse button and choose **Hide** to hide your toolbar or **Properties** to customize it.



## Dialing the MSI HQ BBS

Click on the **Dialing Directory** button near the top left corner of your Qmodem for Windows 95 screen. A new window will open, containing your **Phonebook**.

The QmodemPro for Windows 95 Phonebook is a list of all the Bulletin Board Systems and Online Services you call regularly.

Qmodem for Windows 95 comes with a sample phonebook. The first entry on the list is the MSI HQ BBS. QmodemPro for Windows 95 also includes phonebook entries for some popular online services.

To select the entry you wish to dial, Mustang Software, move the mouse pointer to the entry, and click the left mouse button. When you do that, your selected entry is highlighted — that is, it changes color on your screen.

Now you're ready to dial the entry you just selected. Move your mouse to the **Dial** button near the top left corner of the Phonebook window, and press the left mouse button. You can also double-click on an entry to dial it.

Another window will pop up, showing the name of the system you're calling and details about the entry. These details include how many times you've called, and a status report which shows the number you dialed, and the response from the modem — ringing, busy, connected, and so on.

If the number you're dialing is busy, Qmodem for Windows 95 will pause for a few seconds and try again. The MSI HQ BBS has over two dozen phone lines, so it should only take a few tries to get through if the line is busy on your first try.

Next, we'll [connect to the BBS](#).

## Connecting

When your modem connects successfully with the modem at the other end, you may hear some beeps and squawks as the modems "handshake" or establish a connection. At this point, the modems are exchanging information about the baud rates and connect speeds they support, along with additional information about error correction protocols.

When the modems have completed their handshaking, the phonebook window and status window will disappear, returning you to the terminal window. At this point, the remote system (in this case the MSI HQ BBS) will begin sending data, which you will see on your screen.

Most bulletin board systems have a **command line** interface. That is, they display a menu of available commands, and the user types the letter highlighted on the menu to execute the command.

Next, [log on](#) to the system.

## Logging on

The first screen of information you see on the screen will normally be a brief message identifying the name of the system you've called. You'll then be asked to **log on**.

Logging on means typing your **user ID** (usually your first and last name) at the prompt asking you for this information. Type your first name at the prompt, and press [ENTER] to continue.

You'll then be asked to enter your last name. Type your last name, and once again press [ENTER] to continue.

The BBS will then search its user database for your user ID. Since you're a new user on the BBS, you won't yet be in the database and the BBS will say something like

```
Your name was not found in the user database.  
Is your name spelled correctly?
```

Answer **Yes** to this question if you spelled your name correctly. The BBS will then create a new user record for you, and you'll be asked to select a password.

Think carefully about the password you select. It should be one you can remember easily, but would be difficult for someone else to guess, since this password is your key to the Bulletin Board System. Do NOT use your first name, your spouse's name, or names or words easily associated with you.

The best passwords are a combination of two unrelated words separated by a punctuation mark, for instance

```
RAINBOW$GARBAGE
```

Later on, we'll show you how to enter your password in the phonebook for each system you call, and recall it instantly with a function key.

Sometimes you may be asked for a password and you've never logged on before. [Click here for a possible solution](#), or [continue](#).

## A common password problem

Some people with common names (like "Bill Williams" or "Jenny Johnson") may find that someone else with the same first and last names has already set up a user account on the BBS. If you logged on with the same name as a user already on the system, you'll be asked to enter your password. Since you obviously don't know the password, the easiest thing to do right now is hang up and try again. Use the **Hang Up** button on the right-hand side of the toolbar (it looks like a little telephone that's off the hook) to do this.

Since each user name must be unique, you might want to consider using first name plus your middle initial and last name to identify you. To log on with a middle initial, enter your name like this, at the "What is your first name" prompt:

```
Bill M;Williams [ENTER]
```

Note the semicolon after your middle initial — this notifies the Bulletin Board System that the text "Bill M" should be treated as the first name, and the text "Williams" should be treated as the last name.

Next, you will be asked for [New User Information](#).

## **New User Information**

As part of the new user logon procedure, the operators of the BBS may ask you for some demographic information — your mailing address, phone number, and your QmodemPro for Windows 95 registration number. You will find this number stamped on the card bound inside the cover of this manual.

MSI (the makers of QmodemPro for Windows 95 and other communication software) uses this information to determine customer needs and preferences when planning future product releases. The information you provide is confidential, and is not made public.

[Now you can explore the BBS.](#)

## Exploring the BBS

When your new user information has been stored, you will see several welcome screens, with information of interest to MSI customers. After you read these screens, the system will look for any personal mail for you. If there is new mail for you, the system will let you know. You can read your mail now or wait until you've been explored the BBS.

When the BBS finds new mail, it will ask you if you want to (R)ead new mail, (L)ist new mail, or (C)ontinue. Press **R** to read any new mail messages.

Next, you will arrive at the main Bulletin Board System menu. This is the point from which you can select all the options the BBS has to offer: file transfers, e-mail, doors to other applications, and more. Take a minute or two to look at your options in the main menu.

You can access all the other sections from here. Press the letter of the section you want to visit. You may want to change your settings. Type "**Y**" to see and change your settings.

You can enter your personal information, like what type of computer you're using, determine the Help Level that's best for you (for now, Novice is probably the best. Later, you may want to change it to Regular), turn on auto spell checking, and many other options.

The next thing we'll do is check for new mail. If you chose not to read your new mail when you first logged on to the BBS, you can read it at any time. To check for personal mail:

From the **BBS Main Menu**, select **Read All New Personal Mail**. Type **!** (an exclamation point) at the prompt. The system will search all the conferences for any new mail for you.

Downloading a file is easy! Let's [continue](#).

## For More Information

If you're interested in learning more about modem communications and the on-line community, we recommend the following publications:

"Using Computer Bulletin Boards", 3rd edition, by John V. Hedtke (1995, MIS Press, ISBN 1-55828-391-9). This book explains in easy to understand terms how to log on to a BBS, how to read and answer e-mail messages, how to transfer files, and much more. Ideal for the beginner who wants practical, non-technical information.

"Modems for Dummies", by Tina Rathbone (1993, IDG Books Worldwide, Inc., ISBN 1-56884-001-2). This book gives you a good overview of modems and telecommunications basics. It explains in plain, easy to understand language how modems work.

"BBSs for Dummies", by Beth Slick and Steve Gerber (1995, IDG Books Worldwide, Inc., ISBN 156884-900-1). An in-depth, easy to understand book, written in plain English, that explains BBSs, cyberspace, and more.

"Boardwatch" magazine, published monthly by Jack Rickard. Call 1-800-933-6038 for subscription information. A witty, well written, sometimes opinionated overview of BBS issues. Many good BBS lists, by region and theme.

Most cities and towns have a PC User Group. These groups serve the non-professional computer user by providing social and educational meetings, special interest groups, informal classes for beginners, and guest speakers from the computer industry. Most also provide a Bulletin Board System (BBS) for their members.

Your local computer store or public library can provide information on PC User Groups in your area. If you work for a large corporation or government agency, your employer may also sponsor a PC User Group — ask your PC support people for more information.

The on-line community is as close as your nearest Bulletin Board System. To help get you started, we've included a text file containing a list of bulletin board systems using Wildcat! BBS software from MSI. When you installed QmodemPro for Windows 95, it added an icon named "BBS List" in your program menu. Click on the icon to view the file.

Besides the Bulletin Board Systems in your own area, there are several national and international commercial on-line services. Your QmodemPro for Windows 95 package may contain informational brochures on how to subscribe to one or more of these on-line services.

## Downloading

The final part of our on-line tutorial will be to **download** (receive) a file. Downloading means copying a file from the remote computer (the host, or BBS, or Online Service) to your computer over the modem, using a file transfer protocol.

Press **F** at the **Main Menu**. The BBS will then show you the **File Menu**. This menu has a list of selections having to do with file transfers. For instance, you can search for a file, list the names and descriptions of files on the system, or view the contents of a file you're interested in.

We have created a special file for this tutorial. It contains a selection of sample scripts and other useful tools for Qmodem for Windows 95 users. The name of the file is Q95INFO.EXE.

The most important thing to remember about transferring a file is that you need to tell the computers at both ends of the connection what you want to do. The **host** computer (the BBS or Online Service) needs to be told first. You do this by selecting **Download** from the file menu — as with our previous examples, type the first letter of the command, in this case **D**.

The BBS will ask you the name of the file you want to download. Type

```
Q95INFO.EXE [ENTER]
```

The BBS gives you the option to select several files, and download them in a batch. Since we are only going to download one file during this session, press your [ENTER] key again when prompted for the second file. The BBS will then ask you if you would like to logoff automatically when the download has finished. Since we're just about finished with this tutorial, answer **Yes** to this question.

Zmodem, the most commonly used protocol, is used automatically for new users of the MSI HQ BBS. It is also the default protocol used for QmodemPro for Windows 95 phonebook entries. The BBS will begin sending the file, and QmodemPro for Windows 95 will begin receiving it automatically. Below is a picture of the File Transfer Status window, showing information about the file transfer in progress.

We will explain more about each part of this window later on in this book. When the file transfer is complete, the download status window will disappear, and you will once again see the terminal screen.

Since you selected **Logoff after transfer** before you began downloading your file, the BBS will now display a 10-second countdown. This countdown gives you the opportunity to change your mind and stay on-line by pressing a key. For now, let the countdown finish. The BBS will automatically log you off and hang up the phone.

[Expand your new file.](#)



## Expanding your file

The file you just downloaded is a self-extracting archive — that is, it acts as a container for smaller files inside, which unpack automatically when you execute the file. To remove the files from their container, click on the **Explore Download** button on the macrobar. This macro launches Explorer in your download directory.

Drag the file from the **Download** directory to the **Scripts** directory. It will be in the QmodemPro for Windows 95 download directory (C:\PROGRAM FILES\MUSTANG SOFTWARE\QMODEMPRO\DOWNLOAD, if you have used the default settings).

Before closing Explorer, double-click on the QMWIN.EXE file in the SCRIPTS directory to extract the files from their container. Then close Explorer.

Next, from the QmodemPro for Windows 95 main menu, select **Scripts/Execute**. A dialog box will appear, asking you to choose a file. The new script files should appear in the list of files. Double click on a filename, or mark the filename and then click Open.

Use Explorer to delete Q95INFO.EXE. The script file you downloaded will run without this file, once it has been extracted from the container.

Now that we've completed the tutorial, it's time to shut down QmodemPro for Windows 95. Use your mouse to go to the **File** menu, and select **Exit**, or click the (close) button in the top right- corner of the screen.

Click [here](#) for more information about modems, BBSs, and the electronic community.

## Troubleshooting Guide

Sometimes QmodemPro for Windows 95 may not work for you the way you believe it should. If your troubleshooting efforts are to have any meaning, do not change more than one software setting at a time because you won't be able to tell which new setting actually solved the problem. If you believe the modem is not working properly, check all connections first. Be sure the telephone line is plugged into the wall outlet and the modem. Always turn **off** the power before opening your PC! If the modem was turned off when you turned your computer on, you may need to reboot with the modem turned on in order for Windows 95 to detect it.

If your modem still doesn't work the way it should after you've verified that it is connected properly, refer to your Windows 95 User Guide or the modem manufacturer's documentation for help with modem problems.

Here are some other possible causes:

[Line Noise](#)

[Flow Control](#)

[Serial Ports](#)

[Software Settings](#)

[Transfer Problems](#)

[GP Faults](#)

[Lockups](#)

[Call Waiting](#)

**Line Noise**

Are your telephone lines exceptionally noisy? Noise over the phone line 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.

Did someone pick up an extension phone in your home, while you were on-line? This frequently causes data calls to disconnect abruptly.

## Flow Control

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). Windows 95 defaults to Hardware Flow control. If you wish to change this setting, it can be changed either in the QmodemPro for Windows 95 phonebook properties settings (through **Edit/Entry/Properties/ Advanced**) or in the Windows 95 Control Panel.

## Serial Ports

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.

One possible way to correct this might be to check the FIFO settings in **Modems/Port Settings/Advanced** in the Windows 95 Control Panel. This may data to flow into the processor more rapidly. Refer to the Windows 95 manual for more information about this setting.

## Software settings

Software settings are another possible cause for garbled screens. There are a couple of settings in the QmodemPro for Windows 95 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). These properties must be changed in Windows 95 **Settings/Control Panel/Modems**.

Another possibility is that you've got **8th bit strip** turned **on**. When you're connected to a system that is doing this, select **Connection/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 95 is thus not interpreting the ANSI escape sequences (a lot of [ brackets and numbers) as graphics and color. Try changing the emulation by selecting **Connection/Emulations** and select ANSI.

You should suspect an error in the way you have set up QmodemPro for Windows 95 if you have the same problems on every remote host you call.

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.

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, 95 and if the modem itself if it has flow control settings, check them. Refer to your modem manufacturer's documentation.

Another thing that you might check is the modem properties in the Windows 95 **Settings/Control Panel/Modems**

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.

## **File Transfer Problems**

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.

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 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 Windows 95 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.

If you have a problem, check the system properties in the Windows 95 Control Panel. There could be a conflict causing one of the ports to be disabled. Refer to your Windows 95 documentation, and the manual for your modem, serial card and communication.

If you make changes to the setup, you should change only one thing at a time. This allows you to determine exactly what needs to be changed, and if you make a change that doesn't work, it can be easily corrected.

**GP Fault**

A General Protection or GP fault is only caused by a break or fault in Windows 95, by a driver failure, or a failure in a currently running application, like QmodemPro for Windows 95.



## **Lockup**

If your computer seems to have locked up, press CNTRL+ALT+ DEL. A menu pops up showing all programs currently running. If a program, such as QmodemPro for Windows 95, has locked up or become busy, there will be a message next to that program telling you that the program is not responding. Highlight that program and click the **End Task** button. You will lose any unsaved work in the program, but ending the task will free the rest of the system.

If you are able to make QmodemPro for Windows 95 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.

## Call Waiting

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. You can disable this option by editing either **Edit/Phonebook Entry/Dialing Properties** in the QmodemPro for Windows 95 phonebook or the Windows 95 Control Panel settings. Check the box to toggle the disable option **on**, and choose the appropriate number to dial to disable it.

## A note on conversions

Some of the features in QmodemPro for Windows 95 are completely new. QmodemPro for Windows 95 is the most versatile version yet.

You may be familiar with fax features in older versions of Qmodem. Because TAPI allows programs to "share" modems, Windows 95 includes a fax software package that can share resources with other applications. Now, you can send a fax with any information from QmodemPro for Windows 95 by using the **Send** command. Fax options in QmodemPro for Windows 95 have been removed.

Also, there are no devices in QmodemPro for Windows 95. Windows 95 takes care of all the device (now called modem) information. You are still able to fine tune your modem in QmodemPro for Windows 95, but in most cases, you won't need to.

If for some reason you must "talk" directly to your modem, you now must tell QmodemPro for Windows 95 and Windows 95. To issue commands directly to your modem, you must first select the [Direct Connection](#) option.

You may find that your scripts need to be adjusted and recompiled to remove device, fax, or modem commands that are no longer used by QmodemPro for Windows 95.

If you have used the DOS version of QmodemPro, Qmodem v5, Qmodem Test-Drive, or the older shareware version of Qmodem, you will find QmodemPro for Windows 95 is substantially different.

## The Main Menu

When you first open QmodemPro for Windows 95, you will see the main menu. This is where the menu commands, toolbar, macrobar, statusbar, and terminal screen are launched from.

Menu commands are accessed through this menu. Click on the menu command to execute it, or bring up a dialog box if more information is required. Toolbar commands can be accessed from the toolbar, which is completely configurable, as is the macrobar. Refer to your manual for more information about configuring these options.

The menu command line offers you options for [Files](#), [Edit](#), [View](#), [Connection](#), [Scripts](#), and [Tools](#). It also gives you access to help.

The [terminal screen](#) is where all of your online data will normally appear.

The macrobar appears at the bottom of the screen. You can have a different macrobar for each phonebook entry, if you want.

Under the macrobar is the statusbar. This shows you the current state of QmodemPro for Windows 95.

## Customizing the Desktop

You can customize the look of your desktop by using the [Tools/Options/Desktop](#) menu choice.

The desktop pattern is a simple repeating geometric pattern made of two color 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 [Color](#) options allow you to change the colors that make up the pattern. Use the foreground and background color buttons to view a list of available color options.

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.

If you have a sound card in your PC, QmodemPro for Windows 95 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 **Tools/Options** menu and select the [Sounds](#) property sheet to make changes. A property sheet will pop up showing a list of sound events you can modify. Highlight the entry you want to edit, and click the **Name/Browse** button. A standard Windows Common Dialog Box will pop up showing a list of WAV files in the bottom window, and options to search different drives and directories in the top window.

This is the drive and directory path *QmodemPro for Windows 95* places files downloaded from a host system. Files can be placed in a different directory when the default path is changed, or moved to a different directory at any time.

This is the path that *QmodemPro for Windows 95* checks first for files when you request an upload. Files can be placed in a different directory when the default path is changed, or moved to a different directory at any time.

This is the path *QmodemPro for Windows 95* checks for script files to compile or execute.



This is the path where *QmodemPro for Windows 95* checks for RIP icons. The default path is C:\PROGRAM FILES\MUSTANG SOFTWARE\QMODEMPRO\RIPICONS. This path only needs to be entered if RIPscrip is used as the default emulation for the phonebook entry.

This is the filename and path assigned to the screen trap (or snapshot) file. Using the Edit/Screen to File command tells QmodemPro for Windows 95 to place the contents of the current screen to this file.

This is the filename and path assigned to the capture file. Using the Edit/Screen to Capture command tells QmodemPro for Windows 95 to place the contents of the current screen capture to this file.

This is the path and filename assigned to be used as the default the session log name. The log file is the repository of information about on line sessions.

This is the default filename assigned to the rollback buffers.

Sets the default print device. This can be a printer port or a file on disk.

When this option is on, *QmodemPro for Windows 95* automatically records activity in the log file (session log). This option can also be turned on manually.

When this option is on, time and date headers (stamped either by date the file was received or the date the file was created) are inserted before data in the Screen Trap, Capture, and Scrollback files.



Enter the name of the phonebook file to use as your default phonebook. The default phonebook is the one that will open automatically when you click on the phonebook icon. Use the browse button to see other filenames, paths, and drives to choose from.

Type in the area code you are calling from. If you are dialing through a PBX or similar system, turn the "Use Country Code and Area Code to dial" feature off.

The name of the country you are calling from should appear here. The default country is the United States. To see other countries, click the mouse on the down arrow to the right of the Country Code window.

Sets the number of seconds to wait between unsuccessful dialing attempts. For instance, if you want to wait 5 seconds before trying again after a timeout or busy response, enter a "5" in the text box.

Sets the default modem to be used when dialing from this phonebook. For a list of modems detected by Windows 95, click on the down arrow to the right of the Dial Modem window.

If this option is turned on, the default phonebook selected above will be loaded when *QmodemPro for Windows 95* is first started. If it is turned off, no phonebook will be automatically loaded.

If this option is turned on, changes made to the phonebook during use will be saved automatically. Otherwise, you will be prompted to save changes any time you close the phonebook after changes are made.

If you select a pattern, the selected pattern is displayed in the terminal window, on the screen area outside the borders of the terminal screen. To see a list of available patterns, click on the down arrow to the right of the text box. A drop down box will appear with a list of available patterns.



Select the color to use for the terminal screen foreground. Click on the down arrow to see all foreground color choices.

Select the color to use for the terminal screen foreground. Click on the down arrow to see all background color choices.

Type in the name of a bitmap (.BMP) file to be displayed on top of any desktop pattern you may have selected. The default wallpaper is none. Choose Browse to see other drives, directories, and filenames to choose from.

When this option is on, no wallpaper is displayed.

When this option is on, the bitmap file is repeated across the screen.

When this option is on, the bitmap file is displayed at the center of the screen.

Scroll mode displays the terminal screen by scrolling upwards as new data is received. The older data will scroll off the top of the screen as new data is displayed.

This requires a choice between two options. Click on each individual option for more detail.



When the Selected Font option is on, you must manually select a font size in the Emulations/Options section. This will not change when the terminal window is resized. Select "Based on Window" if you want the font size to automatically adjust to the screen size.

This option determines the size of the font to be used by the size of the terminal screen. If you resize the terminal window, the size of the font will automatically adjust. Select "Selected Font" if you do not want the font size to change when the screen is resized.

Controls how *QmodemPro for Windows 95* updates the screen to display incoming data. Character mode refreshes the screen each time a new character is received. This is the slowest method . This option should be used if your hardware is not capable of supporting the faster modes.

If this option is turned on, the terminal will accept international keystrokes.

If this option is turned on, a confirmation request will pop up when a hangup command is sent, confirming that you do want to break the current connection. You then have the option to accept or cancel the hangup command.

When this option is selected, toolbar buttons appear in large mode. This means only that they are larger than the normal size icons used as the default size in the toolbar.

Some modems always default to reporting Carrier Detect until told to report the true state of the carrier. If your modem *cannot* be set to report the true state of the carrier detection, this option should be turned on.

This window lists all events, such as connections, busy signals, download successes, etc., allowing you to select .WAV sounds to notify you of their occurrence.



Lists all available sounds and their associated .WAV file. Click the down arrow to display a list of sounds available in the default path. Select Browse to choose from other drives and directories.

Clicking this button lets you preview the sound that you have associated with an event, so that you can change it before you save, if needed.

Lists all emulations available. To set an emulation as default, highlight the emulation and press the Set Default button. To change the properties of an emulation, highlight the emulation and click on Properties.

Allows you to select options for a specific emulation. ANSI, the default emulation, is fine for most users.

Controls what color is selected when Esc[m or Esc]0m is executed. When on, color becomes white on black.

If on, cursor will automatically move to the next line down when writing a character in the last column on the screen. If off, then the cursor does not move down to the next line after reaching the last column.

Controls how the NULL character is handled. If on, NULL character is displayed as a space. If off, Null character is stripped from the incoming screen.

If on, character to the left of cursor is erased when a backspace (^H) character is received. If off, cursor is moved to the left but character remains.



This option adds a line feed character whenever a carriage return is sent to end a line and move to the beginning of the next one. Some remote systems may expect a line feed with each carriage return to advise the system that the end of the line has been reached.

Toggles the stripping of the high bit on and off. If on, the high bit is set by the receiving system.

This setting determines the height of your screen. Changing the value makes the screen taller or shorter.

Sets the number of lines available to be saved in the scrollbar buffer.

Selects 80 character columns per line.

These options allow you to determine whether to respond to an ENQ from the host system. Enable allows a response, Disable allows no response, and CIS B+ sends the response commonly requested by CIS B+ emulation. CIS+ B transfers are started automatically and controlled by the host.

If Enable is selected, type in Enquire Sequence that is expected by the host system. If enable is not selected, any entry In this field will be ignored.

Opens the emulation keyboard mapping dialog box, allowing you to attach commands to function keys and other keyboard keys.



Restores the selected emulation to its default settings, undoing any changes in the setting you may have made.

Select the font you want the emulation to use. Click on the down arrow to see a list of available fonts for the specified emulation.

Click the down arrow to the right of the list box to see a list of sizes available for the chosen font.

Select the color that you want the text to appear. Click on the down arrow to see a palette of available colors.

Select the color used as the background color for the text. Click on the down arrow to see a palette of available colors.

Toggles the ANSI music option on and off.

When downloading files, this option causes files with the same name as the one being downloaded to be overwritten. If this option is on, the autoincrement option must be off.

When downloading files, automatically increments the last character of the filename when a file with the same filename exists in the download path.



When a download aborts, this option saves the portion of the file that was successfully received.

When toggled on, allows you to view .BMP, .GIF, and .JPG files as they are downloaded. If this option is not chosen, pictures are not automatically viewed during download.

The file date header for batch transfers is set as the actual date and time they were received by your computer.

Sets the file date header for batch transfers as the original date of the file.

Opens a property sheet allowing you to set upload and download options specific to the ASCII protocol.

Opens a property sheet allowing you to set error detection and other settings specific to the Kermit protocol.

Opens a property sheet allowing you to set CRC and Autostart options specific to the Zmodem protocol.

Toggles the QuickLearn option, allowing *QmodemPro for Windows 95* to learn a script by recording online commands.



Select a translation table to be used from the list of available tables.

Opens a dialog box allowing you to add a translation table to the available list. This can be one that you create yourself, but it must reside in the *QmodemPro for Windows 95* directory.

Removes the highlighted translation table from the list of tables available for selection.

Opens a dialog box which allows you to change incoming and outgoing translation of specific characters.

Sets the highlighted table as the default. This table is then shown in the Default box.

Opens a dialog box allowing the customization of the available toolbar buttons.

Opens a dialog box allowing the customization of available macrobar buttons.

Opens a dialog box requesting a text filename and loads it into the Text Editor.



Opens a dialog box requesting a .BMP, .GIF, or .JPG filename and loads it into the Picture Viewer.

Opens a dialog box requesting a script. This command then opens the script and executes it.

Opens a dialog box requesting a script filename., then opens the Script Editor and loads the requested script.

Opens a dialog box requesting a script filename, then compiles the requested script.

Opens a dialog box requesting a script filename, then compiles the script. If the script has errors, the debugger is then opened.

Loads and executes the Host Mode script, allowing you to receive incoming data calls.

Opens a list of emulations available. Select the emulation you want to use, and click OK.

Clears the current terminal screen of any data.



Resets the current emulation to its default settings. If you have made changes to the emulation settings, this command will restore the original default setting, undoing your changes.

Sends a hangup command, forcing the modem to terminate the current connection.

Sends a break command, which is used to tell the receiving computer to interrupt whatever it is doing.

Most systems other than mainframes do not use Xon/Xoff. If Xon/Xoff flow control was enabled erroneously, use this command to clear the XOFF flow control.

Toggles the Full Duplex option on and off. When on, the host system echoes your keystrokes back to you. A check mark appears next to this when the option is turned on.

Toggles the "Add LF to CR" off and on. When on, a checkmark appears beside the option, adding a line feed to carriage returns.

Toggles the 8th bit strip command on and off. When on, a checkmark appears beside the option.

Allows you to send the entire PC keyboard scan codes to the remote system. The remote system must support this feature for it to work properly. A checkmark appears beside this menu selection if this option is turned on.



Toggles the Auto Answer option on and off. When on, your system is set to receive incoming data calls. A checkmark appears by this option when it is turned on.

Allow a direct connection to your terminal, bypassing TAPI and your modem.

Allows you to choose the port to connect directly through from a list of available ports detected by Windows 95.

Specifies the baud rate for this port. If you are unsure what the value should be, consult your hardware documentation or use the default value. Default is 19200.

Specifies the parity for this port. If you are unsure what the value should be, consult your hardware documentation or use the default value. Default is None.

Specifies the data bits for this port. If you are unsure what the value should be, consult your hardware documentation or use the default value. Default is 8.

Specifies the stop bits for this port. If you are unsure what the value should be, consult your hardware documentation or use the default value. Default is 1.

Specifies the baud rate for this port. If you are unsure what the value should be, consult your hardware documentation or use the default value. Default is 19200.



Specifies the data bits for this port. If you are unsure what the value should be, consult your hardware documentation or use the default value. Default is 8.

Specifies the parity for this port. If you are unsure what the value should be, consult your hardware documentation or use the default value. Default is None.

Specifies the stop bits for this port. If you are unsure what the value should be, consult your hardware documentation or use the default value. Default is 1.

Specifies the flow control for this port. If you are unsure what the value should be, consult your hardware documentation or use the default value. Default is None.

Restores the settings of a direct connection to the default settings (19200, 8, N, 1).

A 16550 UART chip is required for this option, which allows the FIFO buffers to be activated, and the settings used for them.

Toggles the View Toolbar option on and off. If this option is on, a checkmark appears beside the command and the toolbar is visible. If off, the toolbar is hidden.

Toggles the View Macrobar option on and off. If this option is on, a checkmark appears beside the command and the macrobar is visible. If off, the macrobar is hidden.



Toggles the View Status option on and off. If this option is on, a checkmark appears beside the command and the statusbar is visible. If off, the statusbar is hidden.

Toggles the split screen mode on and off. When on, a checkmark appears beside the command, and the screen is divided into two areas, separating incoming (in the top screen) and outgoing text (in the bottom screen).

Opens or closes the File Clipboard. When toggled on, a checkmark appears beside the command, and the File Clipboard is opened, allowing you to save filenames to a list for uploading or downloading.

Allows you to view the scrollbar log. A slider appears along the right side of the terminal window, giving you access to the scrollbar log data.

Toggles the Scrollback Record feature on and off. When on, a checkmark is visible beside the command, and all data from the terminal screen is recorded to the scrollback log. The log can be viewed by using the Scrollback command.

Removes all data from the rollback log.

Copies selected text to the clipboard. The CTRL+INSERT key combination can be used for this function, also.

Pastes selected text from the clipboard. The SHIFT+INSERT key combination can be used for this function, also.



Opens an option allowing you to send the current screen data to the clipboard, a capture file, a trap screen, the printer, or a mail message.

Sends the current screen to the Windows clipboard, where it can be pasted into other screens or applications. Using this command will cause existing clipboard data to be deleted.

Opens a dialog box allowing you to select a capture file filename and send the current screen information to it. You can accept the default capture file name, or select a different drive and directory file.

Opens a dialog box allowing you to select a trap screen filename and send the current screen information to it. You can accept the default trap screen filename, or select a different drive and directory file.

Prints the current screen to the currently selected printer.

Opens the Microsoft Exchange and places the current screen into to an electronic mail message.

Opens an option allowing you to send the scrollback log to the clipboard a capture file, a trap file, the printer, or an electronic mail message.

Sends the scrollbar log to the Windows clipboard, where it can be pasted into other screens or applications. Using this command will cause existing clipboard data to be deleted.



Opens a dialog box allowing you to select a capture file filename and send the scrollback log to it. You can accept the default capture file name, or select a different drive and directory file.

Opens a dialog box allowing you to select a trap screen filename and send the scrollback log to it. You can accept the default trap screen filename, or select a different drive and directory file.

Prints the scrollback log to the currently selected printer.

Opens the Microsoft Exchange and places the scrollback log into to an electronic mail message.

Opens a dialog box allowing you to choose selected text to search for. Search by whole word or case, and up or down from the cursor.

Quits (and closes) *QmodemPro for Windows 95*

When the printer is toggled on, data is sent to the currently selected printer as well as to the terminal. When this option is on, a checkmark appears beside the selection.

Opens a dialog box allowing you to select a saved log file and loads it into the text editor for viewing.



Starts or stops the Session log, saving it to a default or defined filename . This log is used to keep track of major *QmodemPro for Windows 95* activity. When this option is on, a checkmark appears beside the menu selection.

Opens a dialog box prompting you to accept the default filename to save to or modify the path and filename. This command toggles the capture log on and off. When on, a checkmark appears beside the menu option, and all data is sent to a capture file as well as to the terminal screen.

Opens a dialog box prompting you to select a download protocol, directory, and then begins to download a requested file from the remote system.

Opens a dialog box prompting you for a list of files and their paths to send to the remote system, and then lets you send them.

Opens the Dialing Directory, or Phonebook, where dialing information and commands are kept.

Brings up a dialog box letting you choose options to customize the toolbar by adding, removing, and rearranging buttons.

Hides the toolbar. The toolbar can be viewed by toggling this option to Show Toolbar.

Hides the macrobar. The macrobar can be viewed by toggling this option to Show Macrobar.



Opens a dialog box allowing you to map programmable keys, save, and load other macrobars.

Hides the statusbar. The statusbar can be viewed by toggling this option to Show Statusbar.

Holding down the mouse on the titlebar allows you to change the location of the current window.

Allows you to change the size of the current window.

Reduces the size of the current window to an icon.

Enlarges the window to full size.

Closes the current window and prompts save information.

Clicking the right mouse button on the titlebar drops down a list of the following commands:

[Restore](#)IDH\_Rightmouse\_Titlebar\_Restore

[Move](#)IDH\_RightMouse\_Titlebar\_Move

[Size](#)IDH\_RightMouse\_Titlebar\_Size

[Minimize](#)IDH\_Rightmouse\_Titlebar\_Minimize

[Maximize](#)IDH\_RightMouse\_Titlebar\_Maximize

[Close](#)IDH\_Rightmouse\_Titlebar\_Close



Toggles the toolbar view on and off. When the Toolbar View is on, a checkmark appears beside the menu item and the toolbar is visible on the screen.

Toggles the statusbar view on and off. When the Statusbar View is on, a checkmark appears beside the menu item and the statusbar is visible on the screen.

Displays phonebook entries by using large icons. In this view, only the large icon assigned to the phonebook entry and the entry name are visible.

Displays phonebook entries by using small icons and the entry name, displayed across the phonebook window. In this view, only the icon assigned to the phonebook entry and the entry name are visible.

Displays phonebook entries by listing a small icon and entry name. This is similar to the Small Icon view.

Displays phonebook entries in detail view. All the user information about the phonebook entry is visible service (system name), primary number, modem, ID, etc.

Allows you to arrange the icons in your phonebook by name, number, last call date, total calls made, or automatically by *QmodemPro for Windows 95*.

Arranges the entries in the current phonebook alphabetically by system name.



Arranges the entries in the current phonebook numerically by primary number.

Arranges the entries in the current phonebook by last call made. The most recently called entry will be placed at the end of the entry list.

Arranges entries in the current phonebook in the order of total calls made to each entry.

Lets *QmodemPro for Windows 95* automatically line up phonebook entries in the order they appear.

Opens a dialog box prompting you for information about the number to be dialed, then dials the number.

Opens the dialer and dials the selected phonebook entry. Dialing status information is displayed until a connection is made.

Opens the Add Phonebook Entry property sheet, allowing you to create a new entry in the current phonebook.

Opens the Add Phonebook Entry property sheet and then inserts the new phonebook entry at the current location.



When a phonebook entry is highlighted, this command deletes the currently selected phonebook entry. You will be prompted for confirmation before deletion of the entry.

Opens the Edit Phonebook Entry property sheet for the selected phonebook entry, allowing you to view and change the properties of that entry.

Opens the Edit Groups dialog box. This menu option lets you add, delete, and edit phonebook entry groups.

Opens a Phonebook Conversion Wizard which will guide you step-by-step in converting phonebooks from other formats into the *QmodemPro for Windows 95* format.

Creates a new phonebook. You can copy existing entries from another phonebook to the new one, create new groups, add or revise entries. The new phonebook cannot have the same name as an existing phonebook.

The Connection property sheet contains text boxes allowing you to change the properties of the selected phonebook entry. These include [system name](#), icon, [phone number](#), [country code](#), [dialing properties](#), [alternate numbers](#), [modem](#) and [type](#).

Type the name of the system to be called in the text box. This name will be the one that appears in the phonebook. If you want to change the icon associated with the entry, click the Change Icon button.

Type the area code of the system to be dialed. This is a required field, even if it is a local call. TAPI will not use the area code to dial unless it is needed.



Type in the primary number to be used to dial the system. Do not use alpha characters, because TAPI will not recognize them.

Select the country that the system you are calling is located in. The default setting is the United States of America. To see a list of other countries recognized by Windows 95, click the down arrow at the right of the text box. A list will drop down. Click on the country you want to select it.

Opens a property sheet allowing you to set the properties for your own dialing options, including your default location, area code, country, special commands for your dialing system, and credit card dialing.

There are four text boxes allowing alternate numbers to be entered if the system you are calling has more than one phone number available. Note: All the alternate numbers must have the same area code and country code as the primary. If not, a separate entry should be created for each alternate number.

Choose the modem to use for dialing the entry. If you do not want to use the default modem, click on the down arrow at the right of the text box to see a drop down list of available modems seen by Windows 95. Click on the desired modem to select it.

Select the type of connection you want to make. This can be Data, Voice, or Telnet. Click the down arrow at the right of the text box to make your selection. Choose the type of communication you want to use for the phonebook entry. If you will be calling a BBS, an online service, or another type of data-based connection, choose **Data**. If you want to use the QmodemPro for Windows 95 phonebook to dial a voice line, select **Voice** type. One possible advantage of using QmodemPro for Windows 95 to dial for you is the automatic display dialog box, which shows you the length of your call, allows you to hang up, and can log all the information to a log file. The **Telnet** connection type should be selected exclusively for telnet connections.

Opens a property sheet allowing you to change your configuration properties for this modem. Connection control, dial control, status control, connection preferences, call preferences, port settings, and general preferences options are available.

The Options sheet in Edit Phonebook Entry allows you to enter information about the [emulation](#), [transfer protocol](#), [User ID](#), [password](#), [notes](#), [macros](#), [script](#), [RIP icons](#), [dial retries](#) and [translation table](#) you want to use for the phonebook entry.



Select the emulation you want to use for this phonebook entry. Click the down arrow to the right of the text box to see a list of available entries. Click on the emulation you want to use to select it.

Select the transfer protocol (used for downloading and uploading) emulation you want to use for this phonebook entry. Click the down arrow to the right of the text box to see a list of available entries. Click on the protocol you want to use to select it. The default protocol is Zmodem, which is recommended for most users.

Type in the name that you use when connecting to this system. This item can also be used in scripts by using the USERID function.

Type in the password you use for this system. This item can also be used in scripts by using the PASSWORD function.

Type in the name of any note file you want associated with this entry. A note file can be any ASCII text file. The file can contain notes about the current phonebook entry, and *QmodemPro for Windows 95* will search this note file during a text search. Use the browse button to select a note file from a different directory or drive.

Enter the name and path of any macro file you want associated with this phonebook entry. When the entry is dialed, the macros from this file will be displayed on the macrobar at the bottom of the terminal window. Choose the browse button to select a macro file from a different drive or directory.

Type in the name of the script file you want associated with this entry. If you enter a script file in this field, it will be executed automatically when *QmodemPro for Windows 95* connects with the system in this entry, when the dial command is started from the phonebook. Choose the browse button to select a script file from a different drive or directory.

If you are using the RIPScrip emulation for this entry, use the browse button to select the path, directory, and filename where the RIPScrip icons are stored, if it is different from the default. If you enter a path here, *QmodemPro for Windows 95* checks it first, then the system path.



Enter the number of times you want the system to attempt to dial this entry until a connection is established. A zero in this field indicates an infinite number of retries.

Enter the name of any translation table you want associated with this phonebook entry. When the entry is dialed, the translation table will be loaded in and used. Choose the browse button to select from available translation tables.

This is the last date a connection was made to this phonebook entry.

This is the last time a connection was made to this phonebook entry.

The number of times you have successfully uploaded a file to this phonebook entry.

The number of downloads successfully received from this phonebook entry.

The number of times that you have connected successfully to this phonebook entry.

Click this button to clear all the statistic for this phonebook entry.



This is the main Edit Groups dialog box. From here, you can add new groups, delete groups, change the properties of the group, tag all the group entries in the phonebook, or dial the group entries.

Opens the Edit Group dialog box, allowing you to create a new group

Deletes the selected group. You are prompted for confirmation before the deletion occurs.

Opens the Edit Group dialog box, letting you change the properties of the group, such as the group name, entries selected for the group, and the dial mode.

Highlights all group members in the current phonebook.

Dials all members of the group

Closes the dialog box. You will be prompted to save any changes.

Allows you to change the groups name, dial mode, and associated phonebook entries.



The name of the selected group appears in the text box. To change the group name, type the new name in the text box.

If Continuous for All is checked, each entry in the group will be dialed until every entry has achieved a successful connection. If this option is not used, the dialer will stop after the first successful connection made to any group entry.

Shows all the statistics associated with the current phonebook entry, including [last date](#) on, [last time](#), number of [uploads](#) and [downloads](#), and times [connected](#). Pressing the Clear button will reset all the statistics to zero.

Allows you to enter information for a manual dial. This information is sent as if it were a phonebook entry, but is not checked for accuracy. The information needed for this dialog box includes [area code](#) and [number, country code, connect using, type](#) of connection to establish, and [dialing properties](#), if needed.

Type the area code of the system to be dialed. This is a required field, even if it is a local call. TAPI will not use the area code to dial unless it is needed.

Enter the phone number you want to dial. Do not use alpha characters. They are not recognized by the TAPI dialer. This number will not be checked, so be sure you type in the correct number.

Opens a Windows 95 property sheet allowing you to set the properties for your own dialing options, including your default location, area code, country, special commands for your dialing system, and credit card dialing.

Choose the modem to use for dialing the entry. If you do not want to use the default modem, click on the down arrow at the right of the text box to see a drop down list of available modems seen by Windows 95. Click on the desired modem to select it.



Opens a Windows 95 property sheet allowing you to change your configuration properties for this modem. Connection control, dial control, status control, connection preferences, call preferences, port settings, and general preferences options are available.

Select the type of connection you want to make. This can be Data, Voice, or Telnet. Click the down arrow at the right of the text box to make your selection.

Click this button to begin dialing the number, once you have entered all the information and are satisfied that it is correct. A progress box will appear, showing you current dialing information.

Click this button to cancel the dial request and close the dialog box. No information will be saved.

Searches for specified text in the phonebook. Text in any attached notes file is also searched. Searches can be made in either direction of the current entry, finding each occurrence of the text, and if requested text is found, entries with text string can be tagged

Type in the text to search for. All phonebook entry fields are searched, and the search is case insensitive.

Continues the search for text, using text that is currently in the Text to find window.

Tags all phonebook entries with text matching the requested search criteria.



If Forward is selected, search is forward from the cursor to the end of the file. If Backward is selected, text is searched for from the cursor towards the beginning of the file.

Closes the dialog box. Note: The text is not removed, and appears the next time the Find Phonebook Entry dialog box is opened or until changed or removed.

Lists all entries in the current phonebook. To add an entry to the group, highlight the entry and click on Add.

Shows all phonebook entries added to the current group.

Clicking on this button copies all highlighted entries from the list of phonebook entries and places them into the Selected for Group text box.

Removes highlighted entries from the Selected for Group list. The phonebook entry still appears as a phonebook entry, and can be re-entered into the group at any time.

Clears all entries from the Selected from Group text box. All entries you want to put in this group must then be re-added.

Closes this dialog box and saves any changes you have made.



Closes this dialog box without saving any changes you have made.

There are two text boxes. Dragging a filename into the upper one allows you to edit the filename, making any changes needed. The lower text box is where the filenames are "stored" in a list.

Adds an edited filename from the upper text box, where it can be edited, to the lower text box, where it can be saved.

Removes the highlighted filename from the list.

Opens a Windows Common Dialog box allowing you to browse for a file to load into the file clipboard for download or editing. Files saved in the file clipboard have a .QFL extension.

Opens a Windows Common Dialog Box, allowing you to save the list as a different filename. This name should indicate what the files might have in common. This command will not overwrite an existing file.

Searches the terminal screen from the cursor upward toward the beginning of the list for highlighted text.

Searches the text on the terminal screen from the cursor downward for highlighted text.



Select the character used to separate filenames. Click on the down arrow at the right of the text box to make a selection. Options available are Carriage Return, Line Feed, Comma, and Space.

If this is on, long filenames are accepted, and sent. Some systems do not accept the long filename structure, so be sure that the remote system does before you enable this option.

Deletes each filename from the current list as soon as it is sent to the terminal window.

Issues the command to send the list of filenames to the terminal windows. Launches the download.

Filenames can be dragged into a text box, where they can be stored. Allows you to cut, paste, search and send filenames from the terminal window. Save groups of filenames to be downloaded at any time. Closes this property sheet and saves any changes you have made.

Closes this property sheet without saving any changes you have made.

This dialog box makes all options available to customize your toolbar, [add](#), [remove](#), and [change the order](#) of toolbar [icons](#) shown on the toolbar.

This scroll-down list shows all the icons available and their function. Any of these icons may be added to the toolbar by highlighting it and clicking Add.



This is a list of the icons currently visible on the toolbar. These may be removed or added to, and the order may be changed, by highlighting the icon to change and clicking on the appropriate button.

Adds the highlighted button from the Available Buttons box to the Toolbar Buttons box, making it visible on the toolbar.

Removes a highlighted icon currently on the toolbar to the Available Buttons text box. This icon is still available, but will not be displayed on the toolbar unless it is added again.

Moves the selected button up one line. On the toolbar, it moves one space to the left.

Moves the selected button down one line. On the toolbar, it moves one space to the right.

Closes the dialog box, accepting any changes you have made.

Restores the default set of icons in the currently active window.

If you are sending a file to a computer that uses something other than the standard carriage return/line feed indication that the line has ended, you may want to change the handling. Click on the down arrow to the right of the text box to make your selection. Options include Receive Carriage Return, Send Line Feed after Carriage Return, and Strip Carriage Return.



If you are sending a file to a computer that uses something other than the standard carriage return/line feed indication that the line has ended, you may want to change the handling. Click on the down arrow to the right of the text box to make your selection. Options include Receive Line Feed, Add Carriage Return before Line Feed, and Strip Line Feed.

This is the character that the text editor uses as the last character in a line. most often, this is a line feed or carriage return. If your text editor uses a different character to indicate end-of-line, enter its decimal value here.

If the Use Pace Character is turned on, QmodemPro for Windows 95 will wait until that character is received before it sends the next line. Use this option only when you are sure that the remote system always prompts for each line of text with the same character.

Enter the number of milliseconds to pause between sending characters during an ASCII upload, if one is needed.

Enter the number of milliseconds to pause between sending lines during an ASCII upload, if one is needed.

If this is set to on, a checkmark will appear beside the option, and outgoing data is translated according to the translation table for the emulation currently in effect.

If this is set to on, a checkmark will appear beside the option, and QmodemPro for Windows 95 looks for the character entered in the Pace Character field as a signal to send a line of text.

If this is set to on, a checkmark will appear beside the option, and blank lines will be expanded to include at least one blank space. Many BBSs interpret a blank line (without a space) as the end of a message.



If this is set to on, a checkmark will appear beside the option, and text is displayed on the terminal screen during an ASCII upload.

If you are receiving a file from a computer that uses something other than the standard carriage return/line feed indication that the line has ended, you may want to change the handling. Click on the down arrow to the right of the text box to make your selection. Options include Receive Carriage Return, Send Line Feed after Carriage Return, and Strip Carriage Return.

If you are receiving a file from a computer that uses something other than the standard carriage return/line feed indication that the line has ended, you may want to change the handling. Click on the down arrow to the right of the text box to make your selection. Options include Receive Line Feed, Add Carriage Return before Line Feed, and Strip Line Feed.

The timeout value is the amount of time QmodemPro for Windows 95 waits after receiving the last character before it terminates the transfer and saves the file if a standard end-of-file marker is not received.

If this is set to on, a checkmark will appear beside the option, and incoming data is translated according to the translation table for the emulation currently in effect.

If this option is turned on, QmodemPro for Windows 95 ignores the high bit and saves only seven characters. This option might be required if the online service or remote system sends the 8th bit and causes weird-looking foreign language character to appear on your screen instead of regular text.

If this is set to on, a checkmark will appear beside the option, and text is displayed on the terminal screen during an ASCII download.

If this is set to on, a checkmark will appear beside the option, and tab characters will be displayed as 8 spaces. If this option is turned off, a literal tab character (09 decimal) is displayed.



Default is 1024. The default provided is correct for most circumstances, and should not be changed except on the advise of the System Administrator of the host system you plan to call.

Default is 4. The default provided is correct for most circumstances, and should not be changed except on the advise of the System Administrator of the host system you plan to call.

Default is 0. The default provided is correct for most circumstances, and should not be changed except on the advise of the System Administrator of the host system you plan to call.

Default is 0. The default provided is correct for most circumstances, and should not be changed except on the advise of the System Administrator of the host system you plan to call.

Default is 13. The default provided is correct for most circumstances, and should not be changed except on the advise of the System Administrator of the host system you plan to call.

Default is 35. The default provided is correct for most circumstances, and should not be changed except on the advise of the System Administrator of the host system you plan to call.

Default is 126. The default provided is correct for most circumstances, and should not be changed except on the advise of the System Administrator of the host system you plan to call.

Default is 89. The default provided is correct for most circumstances, and should not be changed except on the advise of the System Administrator of the host system you plan to call.



Default is CRC. The default provided is correct for most circumstances, and should not be changed except on the advise of the System Administrator of the host system you plan to call.

Returns the settings to their original default values, undoing any changes you have made.

Applies the new setting to the currently loaded application.

CRC (Cyclical Redundancy Check) is the error checking method used by Zmodem. In cases where CRC type 32 is supported, it will automatically be used. If 32-bit is not supported, 16-bit will be automatically negotiated. Selecting CRC 16 forces the 16-bit CRC to be used.

When Autostart is on, a checkmark appears beside the menu option and Zmodem file transfers start immediately without prompting for a start command to be sent.

The default for this option is on. Unless you are using a PBX or similar system that requires special dialing considerations, this feature should be turned on. TAPI requires an area code and country code when dialing directly through the modem.

Accepts and saves your selections.

Closes the property sheet without saving changes.



Activates the selected features immediately.

Opens the Edit Notes File dialog box, allowing you to browse through directories and drives to select the Notes file to be attached to the entry.

Opens the Select Macro File dialog box, allowing you to browse through directories and drives to select the Macro file to be attached to the entry.

Opens the Edit Script File dialog box, allowing you to browse through directories and drives to select the Script file to be attached to the entry.

Opens the Browse for Folder dialog box, allowing you to browse through directories and drives to select the Riplcon file to be attached to the entry.

Opens the Browse for Folder dialog box, allowing you to browse through directories and drives to select the default download path.

Opens the Browse for Folder dialog box, allowing you to browse through directories and drives to select the default upload path.

Opens the Browse for Folder dialog box, allowing you to browse through directories and drives to select the default script directory.



Opens the Browse for Folder dialog box, allowing you to browse through directories and drives to select the default Riplcon directory.

Opens the Select Screen Trap File dialog box, allowing you to select the default path and filename for the Screen Trap File.

Opens the Select Capture File dialog box, allowing you to select the default path and filename for the Capture File.

Opens the Select Log File dialog box, allowing you to select the default path and filename for the Log File.

Opens the Select Default Phonebook dialog box, allowing you to select the default path and filename for the Phonebook.

Opens the Select Wallpaper dialog box, allowing you to browse through available directories and drives to select your default wallpaper file.

Displays the selected options as they will; appear on your terminal screen.

Opens a Browse For dialog box, allowing you to set the default for .WAV file sounds associated with events.



Opens a property sheet containing details about the sound you have selected.

Previews the sound you have selected.

Turns off the sound preview.

Shows details about the currently selected sound file.

Shows the volume level of the sound currently selected.

Displays the currently selected fonts and colors options as they will appear on the terminal screen.

Selects 132 character columns per line. The remote system must also support this feature for text to format correctly.

Controls how *QmodemPro for Windows 95* updates the screen to display incoming data. Line mode refreshes the screen each time a new line is received. This is the slowest method . This option should be used if your hardware is not capable of supporting the faster modes.



Controls how *QmodemPro for Windows 95* updates the screen to display incoming data. Block mode refreshes the screen only when a block of data is received. This is the fastest method. Your hardware must support

When the Repaint option is chosen, the screen is repainted completely when a new line of data is received.

Overrides the Windows 95 definition of the F10 key and allows you to define it yourself.

Retains the Windows F10 mapping which activates the current Windows menu.

Disables the numlock in the remapping window. If this feature is on, the assigned function is active in the terminal screen.

This is used by the keyboard mapper to toggle the numlock pad of the keyboard between numbers and arrows.

Uses the Tab key to move the current screen focus from one item to the next. This option is only valid for the keyboard mapping dialog box.

Uses the keystroke (first letter of the key you want to remap) to change the focus of the current screen. This option is only valid for the keyboard mapping dialog box.



Select this if you are using a 101 key keyboard. This option is only valid for the keyboard mapping dialog box.

Select this if you are using an 84 key keyboard. This option is only valid for the keyboard mapping dialog box.

Assigns the key shown in the normal key mode.

Assigns the key shown as it will be used in conjunction with the shift key.

Assigns the key shown as it will be used in conjunction with the control key.

Assigns the key shown as it will be used in conjunction with the control and shift keys.

Edits format in character mode.

Edits format in decimal mode.



Edits format in hexadecimal mode.

Disables the ability of *QmodemPro for Windows 95* to respond to an ENQ sequence.

Enables the ability of *QmodemPro for Windows 95* to respond to an ENQ sequence.

Enables *QmodemPro* for *Windows 95* to respond to CIS B+ ENQ sequences.

No help topic is available for this object. Click on a specific item for detailed help about this dialog box.

Opens the Save Phonebook dialog box, allowing you to open an existing phonebook.

Saves the current phonebook with any changes you have made.

Saves the current phonebook with changes you have made with a different name, leaving the original intact.



Opens a conversion wizard, allowing you to Import phonebooks from other formats into QmodemPro for Windows 95 format.

Opens the Print Phonebook dialog box, allowing you to print selected entries.

Opens the Print Setup dialog box, allowing you to select printer options.

Exit the current phonebook. You will be prompted to save any changes.

Lets you choose to send the data on the current screen to a file, mail, or the printer.

Lets you choose to send the data in the scrollback buffer to a file, mail, or the printer.

Lets you send selected text to a file, mail, or the printer.

Allows you to view the properties of the current direct connection.



Opens the Toolbar or Macrobar Customize dialog box.

Opens the Translate Tables options dialog box.

Opens the Options property sheet, allowing you to customize QmodemPro for Windows 95 options.

Provides a list of current Help Topics

Provides important registration, contact and version information about *QmodemPro for Windows 95*.

Drops down a File menu, letting you open the phonebook, select data recording and upload/download options.

Drops down an Edit menu, containing options for editing text on screen.

Drops down a View menu, allowing you to change the look of the terminal screen. Also provides access to the File Clipboard.



Drops down a Connection menu, allowing you to change emulation, connection type, and connection settings.

Drops down a Scripts menu, allowing you to create, edit, debug, compile, and execute scripts.

Drops down a Tools menu, allowing you to select tools and options to customize the look of *QmodemPro for Windows 95*.

Provides Help information about *QmodemPro for Windows 95*.

Shows the status of the current connection.

Number keys (1 through 0) that can be assigned in combination with ALT, ALT+SHIFT, ALT+CTRL, and ALT+SHIFT+CTRL as macro keys.

Assigns the command to a keystroke combinations using ALT+ [KEY]

Assigns the command to a keystroke combinations using ALT+SHIFT+[KEY]



Assigns the command to a keystroke combinations using ALT + CONTROL+ [KEY]

Assigns the command to a keystroke combinations using ALT+SHIFT+CONTROL+[KEY]

This section lets you assign macros using the Alt, Shift, and Control keys in various combinations.

Allows you to choose to save and execute the macro assignments as Character, decimal, or hexadecimal format.

Saves the macro assignment in character format.

Saves the macro assignment in decimal format.

Saves the macro assignment in hexadecimal format.

Shows the currently assigned default macrobar.



Shows the macro label as it will appear on the macrobar.

Opens a New Macrobar dialog box, allowing you to create a new macrobar.

Saves the currently open macrobar with changes with a new name, allowing the original to remain unchanged.

Opens a Load Macrobar dialog box, allowing you to load a previously saved macro file.

Sets the currently open macrobar as the default.

Closes the macro keys dialog box, saving any changes you have made.

Drops down the File Menu, allowing you to open, save, import, and print phonebooks.

Drops down the Edit Menu, allowing you to manipulate phonebook entries.



Drops down the Dial Menu, allowing you to select specific phonebook entries and manual dialing entries.

Drops down the View Menu, allowing you to select options to change the way the phonebook looks.

Opens the *QmodemPro for Windows 95* Help engine.

Opens the Phonebook Open dialog box, allowing you to select an existing phonebook to open.

Save the current phonebook along with any changes you have made.

Opens a Phonebook Save As dialog box, allowing you to save changes you have made to a different phonebook name, leaving the original phonebook intact.

Opens the Print Phonebook dialog box, allowing you to select components of the current phonebook and print them.

Opens the Print Setup dialog box, allowing you to select printer options.



Exits the current phonebook, saving any changes you have made.

Removes selected entries and copies them to the Windows clipboard. Same as keyboard command CTRL+X.

Copies the selected phonebook entries to the Windows clipboard. Same as keyboard command CTRL+C.

Inserts the selected entry (or entries) that you have cut or copied from the Windows clipboard to the current cursor location. Same as keyboard command CTRL+V.

Searches phonebook entries for specified text.

Searches phonebook entries for the next incidence of specified text. Used in conjunction with Find.

Opens the Edit Groups dialog box, allowing you to edit a group or groups of phonebook entries.

Opens the Edit Group dialog box, allowing you to edit entries within a group.



Deletes selected groups from the phonebook.

Tags highlighted members of a group.

Dials all the members of the selected group.

Closes the Edit Group dialog box, saving any changes you have made.

Shows a list of the groups existing in a phonebook.

Lets you arrange phonebook entries by name, number, last time called, total number of calls, or automatically.

Lines up phonebook entry icons in the phonebook. Only valid for large or small icon view.

Opens the *QmodemPro for Windows 95* help engine.



Drops down a list of available protocols that may be used to upload files.

Sends the currently selected files from the list.

Removes the selected files from the list of files to be uploaded.

Opens a Load File List dialog box, allowing you to add another list to the currently selected list of files.

Opens a Save File List dialog box, allowing you to save the list of files to be uploaded.

Shows a list of files selected for upload.

Drops down a list of protocols available for use in downloading files.

Accept your selection.



Close the dialog box without saving your selection.

Scrolls to display the data as it is received from the remote system.

Repaints the entire terminal screen as new data is received from the remote system.

Select the type of display scrolling style you want. Choose between Scroll and Repaint.

The terminal font may be based on the window size or on a selected font. Only one option may be chosen.

Available options determining how the screen is updated. Options are Character, Line, or Block. Only one option may be selected.

Selects the use of a 1 byte error detection method.

Selects the use of a 2 byte error detection method.



Selects the use of CRC (Cyclical Redundancy Check) as the default error detection

Select a general Enquire option. This tells *QmodemPro for Windows 95* how to respond to Enquire requests. Only one of the three options may be selected at a time. Click on the specific option to see more details.

This group box has more than one option. Select a specific option for more information.

Shows the icon currently assigned to the phonebook entry.

Opens a dialog box allowing you to change the icon associated with the phonebook entry.

The alternate number that the dialer will try first if no connection is made with the primary number.

The alternate number that the dialer will try second if no connection is made with the primary and secondary numbers.

The alternate number that the dialer will try third if no connection is made with the primary and previous alternate number.



The alternate number that the dialer will try last if no connection is made using the primary or other alternate numbers.

This is the icon associated with the modem shown in the Connect Using list box.

Renames the selected phonebook entry.

Closes the manual dial dialog box without dialing the requested number. No changes are saved.

Toggles the Use country Code and Area Code feature on and off. If on, a checkmark appears beside the option. Default is on. This can be forced off when using a PBX or other system that will not accept the area code and country code in the dialing string.

Lists all countries recognized by Windows 95. To see available options, click the down arrow to the right of the listbox.

Closes the Edit Groups dialog box, saving any changes.

Closes the Edit Group dialog box without saving changes.



Closes the Edit Group dialog box and saves all changes.

Searches phonebook entries for specified text from the current cursor position upwards towards the top of the list.

Searches phonebook entries for specified text from the current cursor position downwards towards the bottom of the list.

Closes the Find dialog box.

Opens the Desktop Properties properties sheet, allowing you to change the look of the desktop.

Opens the Emulation property sheet for the currently active emulation.

Sends selected text to the terminal.

Closes the Select Dialog Protocol without completing download.



Accepts the protocol and opens the Select Directory dialog box.

Shows the default download directory. This can be accepted or changed.

Opens the Browse for Folder dialog box, allowing you to select files from a different drive and directory.

Closes the directory dialog box, accepts your selection, and begins the download.

Closes the Select Directory dialog box without beginning a download.

Displays the total number of files selected for upload.

Displays the total number of bytes in the files selected for upload.

Closes the file clipboard and saves any changes.



The upper of two text windows in the File Clipboard. Drag the filename into this box to edit it. Once you are satisfied with the edit, click Add to add the selection to the listbox.

Lists all the files currently entered in the File Clipboard list.

Closes the Select Emulation dialog box without saving changes.

Closes the Select Emulation dialog box and accepts the selection.

Opens an option box allowing you to send selected text to the clipboard, a capture file, a trap screen, the printer or a mail message.

Opens a dialog box allowing you to select a capture file filename and send selected text to it. You can accept the default capture file name, or select a different drive and directory file.

Opens a dialog box allowing you to select a trap screen filename and send selected text to it. You can accept the default trap screen filename, or select a different drive and directory file.

Prints the selected text to the currently selected printer.



Opens the Microsoft Exchange and places the selected text into to an electronic mail message.

Sends the selection to the Windows clipboard, where it can be pasted into other screens or applications. Using this command will cause existing clipboard data to be deleted.

Opens a Select Scrollback File dialog box, allowing you to select from a different drive and directory.

Selects the highlighted emulation as the default.

Displays the currently selected default emulation.

Allows you to select the use of true type fonts for RIPscrip emulation.

Enables the Vttx emulation to allow 8 Bit Commands.

Displays the currently selected default translation table to use.



List the available emulations to select from.

Closes the Macro Keys dialog box and saves changes.

Activates the one (1) key to assign a macro key combination.

Activates the two (2) key to assign a macro key combination.

Activates the three (3) key to assign a macro key combination.

Activates the four (4) key to assign a macro key combination.

Activates the five (5) key to assign a macro key combination.

Activates the six (6) key to assign a macro key combination.



Activates the seven (7) key to assign a macro key combination.

Activates the eight (8) key to assign a macro key combination.

Activates the nine (9) key to assign a macro key combination.

Activates the zero (0) key to assign a macro key combination.

Displays the name of the macro as it appears on the Alt (A) macrobar.

Displays the name of the macro as it appears on the ALT+SHIFT (AS) macrobar.

Displays the name of the macro as it appears on the ALT+CONTROL (AC) macrobar.

Displays the name of the macro as it appears on the ALT+SHIFT+CONTROL (ASC) macrobar.



The incoming translation table controls which characters pass unchanged and which are translated to other characters.

The outgoing translation table controls which characters are sent out to a remote system unchanged and which are translated into other characters.

Selects the emulation translation format type as Character. Characters are displayed as text.

Selects the emulation translation format type as Decimal. Characters are displayed as ASCII character numbers.

Selects the emulation translation format type as Hexadecimal. Characters are displayed as ASCII character numbers in base 16 format.

Closes the Emulation Translation Table dialog box and saves changes.

Closes the Emulation Translation Table dialog box without saving changes.

Press the key you wish to map. Available keys are shown in black, while unavailable keys are grayed out.



Select the protocol you want to use for downloading. Click on the protocol to select it.

Copies the current data on the screen to the Windows Clipboard.

Copies selected text to a capture file.

Copies selected text to a file.

Sends selected text to the printer.

Opens the Microsoft Exchange and places the selected text into to an electronic mail message.

Uses the error correction method 32-bit CRC. This is automatic unless 32-bit CRC is not available.

Uses the error correction method 16-bit CRC. Zmodem automatically uses 32-bit CRC if it is available. If necessary, select 16-bit CRC to force Zmodem to use this method.



Closes the script debugger, saving changes.

Toggles the toolbar view on and off. When this feature is on, the toolbar is visible and a checkmark appears beside the selection.

Toggles the statusbar view on and off. When this feature is on, the statusbar is visible and a checkmark appears beside the selection.

Jumps the debugger over the subroutine.

Steps the debugger into the subroutine.

Returns the debugger to the calling subroutine.

Runs the script, allowing you to determine any possible errors.

Stops a script from running.



Toggles the Breakpoint on and off. When on, a checkmark appears beside the command and a red dot appears at the point that the script has produced an error.

Opens a Watch Variable dialog box, allowing you to select variables to look for in the script.

Clears all watch variables from the script.

Arranges the open windows in the debugger into a tiled pattern, with all windows visible.

Arranges open windows in the debugger into a cascaded view, one atop the other.

Arranges icons automatically.

Lists the last files viewed. Double click on the files to reopen.

Opens the QmodemPro for Windows 95 Help system.



The requested script is loaded into this window and is viewable.

Displays variables that have been loaded into the Watch Variable window.

Toggles the toolbar view on and off. When on, the toolbar appears at the top of the Debugger window.

Displays the options to toggle the breakpoint, watch a variable, or clear all watches.

Toggles the statusbar view on and off. When on, the statusbar appears at the bottom of the Debugger window.

Displays the current status of the debugging session.

Displays toolbar icons currently active.

Displays a combobox displaying the selected variable to watch for. Click OK to accept and place the variable into the watch window.



Accepts the selection and closes the Watch Variable dialog box.

Closes the Watch Variable dialog box without saving changes.

Deletes the selected watch variable.

Toggles the Hide Statusbar command on. The statusbar is no longer visible when this command is on.

Displays a dialog box allowing you to customize the toolbar.

Toggles the Hide Toolbar command on. The toolbar is no longer visible when this command is on.

Displays the icon associated with the modem you have selected to use when dialing the number.

Displays the incoming character as it is sent by the remote system.



Type in the character to be seen by your system when the incoming character is received.

Closes the translation table dialog box and saves changes.

Closes the translation table dialog box without saving changes.

Displays the outgoing character as it is sent to the remote system.

Type in the character to be seen by the remote system when the outgoing character is sent.

Closes the translation table dialog box and saves changes.

Closes the translation table dialog box without saving changes.

#### Customize Toolbar

Adds, removes, and rearranges buttons on the toolbars in the Viewer, Standard Send form, Standard Read form, and Find windows.

#### Available Buttons

Displays the buttons that you can add to the toolbar in the active window.

#### Add

Moves the button selected in the Available Buttons list to the Toolbar Buttons list.

#### Remove

Moves the button selected in the Toolbar Buttons list to the Available Buttons list.

#### Toolbar Buttons

Displays the buttons in the order they appear (from left to right) on the toolbar in the active window.

#### Reset

Restores the default set of buttons on the toolbar in the active window.

#### Move Up

Moves the selected button up one line. On the toolbar, the button shifts left one space.

#### Move Down

Moves the selected button down one line. On the toolbar, the button shifts right one space.



Closes the Options dialog box and saves changes.

Closes the options dialog box without saving changes.

Displays the key currently being mapped.

Displays the macro assigned to the key.

Displays the macro assigned to the key when used in conjunction with the Shift key.

Displays the macro assigned to the key when pressed in conjunction with the Control key.

Displays the macro assigned to the key when pressed in conjunction with the Shift and Control keys.

Displays the current status of the F10 key. Options are Menu or Mapped.



Displays the current status of the Numlock key. Options are Mapped and Keypad Toggle.

Displays the current method of changing focus for selection. Options are Tab or Keystroke.

Opens the Options dialog box, allowing you to select keymapping options.

Closes the dialog box and saves changes.

Closes the Print Phonebook box without printing the phonebook

Opens the Print dialog box, allowing you to print the phonebook.

Moves the highlighted entry down one position in Print Fields.

Moves the highlighted entry up one position in the Print Fields list.



Removes the highlighted entry from the Print Fields list.

Adds the highlighted entry to the Print Fields list.

Displays a list of phonebook fields selected for printing.

Displays a list of selectable phonebook fields available for printing.

Displays the current icon group path and filename.

Opens the Change Icon dialog box, allowing you to associate a different icon with the phonebook entry.

Closes the Change Icon dialog box and saves changes.

Closes the Change Icon dialog box without saving changes.



Opens a Change Icon dialog box, allowing you to select from a different drive and directory.

The current icon associated with the phonebook entry is highlighted.

Previews the sound you have selected.

Turns off the sound preview.

Choose the modem you want to answer incoming calls by clicking on your selection.

Closes the Auto Answer modem dialog box and saves changes.

Closes the Auto Answer modem dialog box without saving changes.

Displays the length of time connected to this phonebook entry the last time a connection was established.



